Sun Fire™ 6800/4810/4800/3800 System Controller Command Reference Manual

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 U.S.A.
650-960-1300

Part No. 805-7372-11
April 2001, Revision A

Send comments about this document to: docfeedback@sun.com

Copyright 2001 Sun Microsystems, Inc., 901 San Antonio Road • Palo Alto, CA 94303 USA. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X / Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, OpenBoot, Sun Fire, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.UNIX is a registered trademark in the United States and other countries, exclusively licensed through X / Open Company, Ltd.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2001 Sun Microsystems, Inc., 901 San Antonio Road • Palo Alto, CA 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X / Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, OpenBoot, Sun Fire, ,et Solaris des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. UNIX est une marque enregistree aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X / Open Company Ltd.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Please
Recycle

Contents

Preface xv

1. Introduction to the System Controller 1

Platform Administration 2

Platform Shell 2

Platform Console 2

Domain Administration 3

Domain Shell and Domain Console 3

Accessing the System Controller Main Menu 5

Accessing the System Controller Main Menu Using `telnet` 5

▼ To Access the System Controller Main Menu and Enter the Platform Shell Using `telnet` 5

▼ To Access the System Controller Main Menu and Enter a Domain Shell Using `telnet` 6

Accessing the System Controller Main Menu Using the Serial Port 9

▼ To Access the System Controller Main Menu if You Connected the System Controller Serial Port to an ASCII Terminal 9

▼ To Access the System Controller Main Menu if You Connected the Serial Port to a Network Terminal Server (NTS) 9

▼ To Obtain the Domain Shell Prompt From the Domain Console Prompt 10

System Controller Navigation 10

- To Enter the Domain Console From the Domain Shell If the Domain Is Inactive 13
- To Enter the Domain Shell From the Domain Console 14
- To Get Back to the Domain Console From the Domain Shell 14
- To Enter a Domain From the Platform Shell 15
- To Terminate a Session With `telnet` If You Are Connected to the Ethernet Port 15
- To Terminate a Session With `tip` If You Are Connected to the Serial Port 16

2. System Controller Alphabetical Command Reference 17

System Controller Syntax, Command Names, Command Arguments, and Comments 17

Command Syntax 17

Command Names 18

Command Arguments 18

Device Names 18

Board States 19

Status 20

Comments 20

Clearing an Entry 21

System Controller Command Summary 21

Status 23

Domain States 24

Alphabetical Listing of System Controller Commands 25

addboard 25

break 28

connections 30

console 32

deleteboard 35

disablecomponent 37

disconnect 41

dumpconfig 43

enablecomponent 46

flashupdate 49

help 56

history 58

password 59

poweroff 61

poweron 64

reboot 67

reset 69

restoreconfig 72

resume 74

setdate 75

setdefaults 79

setkeyswitch 81

setupdomain 85

setupplatform 90

showboards 94

showcomponent 98

showdate 101

showdomain 103

showenvironment 107

showkeyswitch 111

showlogs 113

showplatform 115

showsc 120

testboard 121

Glossary 123

Index 125

Figures

FIGURE 1-1 Navigating Between the Platform Shell and the Domain Shell 11

FIGURE 1-2 Navigating Between the Domain Shell, the OpenBoot PROM, and the Solaris Operating Environment 12

FIGURE 1-3 Navigating Between the OpenBoot PROM and the Domain Shell 13

Tables

TABLE 1-1 Maximum Number of Active Domains by System Type and Partition Mode 4

TABLE 2-1 Devices That Can Be Managed in the Platform Shell 18

TABLE 2-2 Devices That Can Be Managed in a Domain 19

TABLE 2-3 Board States 19

TABLE 2-4 Status Field Displayed in the `showboards` Command 20

TABLE 2-5 System Controller Command Summary 21

TABLE 2-6 Status Field Displayed in the `showboards` Command 23

TABLE 2-7 Header Description for CODE EXAMPLE 2-4 31

TABLE 2-8 *component_name* Descriptions for the CPU/Memory Board Used in the `disablecomponent` Command 38

TABLE 2-9 *component_name* Descriptions for the I/O Assembly Used in the `disablecomponent` Command 38

TABLE 2-10 `dumpconfig` Example 45

TABLE 2-11 *component_name* Descriptions for the CPU/Memory Board Used in the `enablecomponent` Command 46

TABLE 2-12 *component_name* Descriptions for the I/O Assembly Used in the `enablecomponent` Command 47

TABLE 2-13 `enablecomponent` Command Example Enabling CPU/Memory Board 4 48

TABLE 2-14 poweron Command Example Showing Powering On CPU/Memory Board 2 66

TABLE 2-15 poweron Command Example Showing Powering On I/O Assembly 7 66

TABLE 2-16 poweron Command Example Showing Powering On All CPU/Memory Boards and I/O Assemblies in Domain A 66

TABLE 2-17 Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time 76

TABLE 2-18 `setdate` Command Example in the Platform Shell 77

TABLE 2-19 `setkeyswitch` Values and Description 82

TABLE 2-20 Results of Changing the Keyswitch From the Current Setting to a New Setting 82

TABLE 2-21 `setkeyswitch on` Example 84

TABLE 2-22 `setkeyswitch off` Example 84

TABLE 2-23 `setkeyswitch off` Example Shutting Down Domain B From the Platform Shell 84

TABLE 2-24 Parameter Values for the `setupdomain` Command 87

TABLE 2-25 `setupplatform` Parameter Values 91

TABLE 2-26 Output Header Definitions for the `showboards -v` Command 96

TABLE 2-27 `showenvironment -p` *part* Descriptions 108

TABLE 2-28 `showenvironment` Output Header Description 108

TABLE 2-29 Definitions of Headings for Status of All Domains in CODE EXAMPLE 2-62 119

TABLE 2-30 `testboard` Example Testing CPU/Memory Board 0 122

Code Samples

CODE EXAMPLE 1-1 Accessing the System Controller With `telnet` and Entering the Platform Shell 6

CODE EXAMPLE 1-2 Accessing a Domain Shell Using the `telnet` Command 7

CODE EXAMPLE 1-3 Accessing a Domain Shell From the Domain Console 8

CODE EXAMPLE 1-4 Accessing a Domain Shell From the Domain Console 10

CODE EXAMPLE 1-5 Obtaining a Domain Shell From the Domain Console 14

CODE EXAMPLE 2-1 `addboard` Example in the Platform Shell 26

CODE EXAMPLE 2-2 `addboard` Example in the Domain Shell 27

CODE EXAMPLE 2-3 `break` Command Example in the Domain Shell 29

CODE EXAMPLE 2-4 `connections` Command for the Platform Shell 31

CODE EXAMPLE 2-5 `console` Example With the `-d` Option and No Password Assigned Accessing Domain B 33

CODE EXAMPLE 2-6 `console` Example Without the `-d` Option and No Password Assigned Accessing Domain B 33

CODE EXAMPLE 2-7 `console` Example With the `-d` Option and A Password Assigned Accessing Domain A 33

CODE EXAMPLE 2-8 `console` Example Without the `-d` Option and A Password Assigned Accessing Domain A 34

CODE EXAMPLE 2-9 `deleteboard` Example Showing Deleting I/O Assembly 7 36

CODE EXAMPLE 2-10 `deleteboard` Example Showing Deleting CPU/Memory Board 3 36

CODE EXAMPLE 2-11 `disablecomponent` Command Example Adding `sb4` to the Blacklists 40

CODE EXAMPLE 2-12 `disablecomponent` Command Example Adding `sb0` CPU Port 3 to the Blacklist 40

CODE EXAMPLE 2-13 `disablecomponent` Command Example Adding `sb2,` CPU Port 3, and Physical Bank 0 to the Blacklists 40

CODE EXAMPLE 2-14 `disablecomponent` Command Example Adding I/O Assembly 9, Port 0 and Bus 1 and Also I/O Assembly 8, I/O Card 2 to the Blacklists 40

CODE EXAMPLE 2-15 Disconnecting a Session From the System Controller Platform Shell 42

CODE EXAMPLE 2-16 Disconnecting a Session From the System Controller Platform Console 42

CODE EXAMPLE 2-17 `disconnect` Example Showing Disconnecting a Session From the Domain A Shell 42

CODE EXAMPLE 2-18 `enablecomponent` Command Example Enabling I/O Assembly 6 and Port 1 48

CODE EXAMPLE 2-19 Example of the `flashupdate` Command Entering Single-User Mode 52

CODE EXAMPLE 2-20 `flashupdate` Example Updating the Active System Controller Board and the System Controller Real Time Operating System 54

CODE EXAMPLE 2-21 `flashupdate` Example Updating System Boards (CPU/Memory Boards and I/O Assemblies) 54

CODE EXAMPLE 2-22 `flashupdate` Example Updating CPU/Memory Board 4 54

CODE EXAMPLE 2-23 `flashupdate` Example Updating I/O Assembly 8 54

CODE EXAMPLE 2-24 `flashupdate` Example Updating the System Controller Real Time Operating System 55

CODE EXAMPLE 2-25 `help` Command Example Displaying Information on the `addboard` Command 57

CODE EXAMPLE 2-26 `help` Command Example Displaying Commands Beginning With the Word `show` 57

CODE EXAMPLE 2-27 `help` Command Example Displaying Commands Beginning With the Letter A 57

CODE EXAMPLE 2-28 `history` Command 58

CODE EXAMPLE 2-29 `password` Command Example With No Password Set 60

CODE EXAMPLE 2-30 `password` Command Example With a Password Already Set 60

CODE EXAMPLE 2-31 `poweroff` Command Example Showing Powering Off CPU/Memory Board 2 63

CODE EXAMPLE 2-32 `poweroff` Command Example Showing Powering Off All Power Supplies and Currently Controllable Boards in the Platform Shell 63

CODE EXAMPLE 2-33 `poweroff` Command Example Showing Powering Off All CPU/Memory Boards and I/O Assemblies in Domain A 63

CODE EXAMPLE 2-34 `poweroff` Command Example Showing Powering Off I/O Assembly 7 63

CODE EXAMPLE 2-35 `reset` Command Example From Domain A 71

CODE EXAMPLE 2-36 `reset` -a Command Example From Domain A 71

CODE EXAMPLE 2-37 `reset` Command Example Resetting `ssc0` From `ssc1` in the Platform Shell 71

CODE EXAMPLE 2-38 `restoreconfig` Example 73

CODE EXAMPLE 2-39 `resume` Command Example in Domain A 74

CODE EXAMPLE 2-40 `setdate -r` Command Example Setting the Date From a Date Host 78

CODE EXAMPLE 2-41 `setdate -t` Command Example Setting the Time Zone to Pacific Standard Time Using the Offset From Greenwich Mean Time 78

CODE EXAMPLE 2-42 `setdate -t` Command Example Setting the Time Zone to Eastern Standard Time Using Time Zone Abbreviations and Also Setting the Date 78

CODE EXAMPLE 2-43 `setdate -t` Command Example Setting the Time Zone to Japan Standard Time Using the Offset From Greenwich Mean Time 78

CODE EXAMPLE 2-44 `setdate -t` Command Example Setting the Time Zone to Eastern Central Time Using Time Zone Abbreviations 78

CODE EXAMPLE 2-45 `setdefaults` Example 80

CODE EXAMPLE 2-46 Variables for the `setupdomain` Command 86

CODE EXAMPLE 2-47 Output From `setupplatform` Command 93

CODE EXAMPLE 2-48 `showboards` Command for the Platform Shell 95

CODE EXAMPLE 2-49 `showboards -v` Command for the Platform Shell 95

CODE EXAMPLE 2-50 `showboards` Command for the Domain Shell 97

CODE EXAMPLE 2-51 `showcomponent sb4` Sample Output 99

CODE EXAMPLE 2-52 `showcomponent ib6` Sample Output 100

CODE EXAMPLE 2-53 `showdate` Command for the Platform Shell 102

CODE EXAMPLE 2-54 `showdate` Command for the Domain Shell 102

CODE EXAMPLE 2-55 `showdomain` Command Example 104

CODE EXAMPLE 2-56 `showdomain -v` Command Example 105

CODE EXAMPLE 2-57 `showdomain -p bootparams` Example Displaying Boot Parameter Information 106

CODE EXAMPLE 2-58 `showenvironment` Platform Shell and Domain Shell Example 110

CODE EXAMPLE 2-59 `showkeyswitch` Example Showing the Keyswitch Set to On 112

CODE EXAMPLE 2-60 `showkeyswitch` Example Showing the Keyswitch Set to Off 112

CODE EXAMPLE 2-61 `showkeyswitch` Example Showing the Keyswitch Set to Standby 112

CODE EXAMPLE 2-62 `showplatform` Output for a Sun Fire 6800 System 116

CODE EXAMPLE 2-63 `showplatform -v` Output for a Sun Fire 6800 System 117

CODE EXAMPLE 2-64 `showsc` Command 120

Preface

This book describes how to use the system controller command line interface, which controls the system functions and provides environmental monitoring plus domain and hardware control. With the system controller software, you can configure the platform and domains, as well as power on and off domains, power supplies, fans, and other components. The system controller also provides a platform console and a domain console.

Before You Read This Book

This book is for system administrators, who have a working knowledge of the Solaris™ operating environment. If you do not have such knowledge, you should first read the Solaris User and System Administrator AnswerBooks and consider UNIX® system administration training.

How This Book Is Organized

This book contains the following chapters:

Chapter 1 introduces you to the system controller. It explains the platform shell and the domain shell. It also describes the responsibilities of the platform administrator and the domain administrator. It provides step-by-step procedures on how to enter the platform shell and the domain shell of the system controller, how to navigate within the system controller, between the system controller and the Solaris operating environment, and between the OpenBoot™ PROM and the system controller. It also explains how to terminate a system controller session.

Chapter 2 explains system controller command syntax, command names, and command arguments. It also provides in tabular form a summary of all of the system controller commands and provides complete descriptions, with syntax, and examples for each system controller command.

Typographic Conventions

Typeface	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files.
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the *User's Guide.* These are called *class* options. You *must* be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type `rm` *filename.*

Shell Prompts

Shell	Prompt
C shell	*machine_name*%
C shell superuser	*machine_name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#
Platform shell	*schostname*`:SC>`
Domain shell	*schostname*`:A>` or `B>`, `C>`, `D>`

Related Documentation

Application	Title	Part Number
Service	*Sun Fire 6800/4810/4800/3800 Systems Service Manual*	805-7363
Service	*Sun Fire 4810/4800/3800 System Cabinet Mounting Guide*	806-6781
System Administration	*Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*	805-7373

Accessing Sun Documentation Online

The `docs.sun.com`[sm] web site enables you to access a select group of Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

```
http://docs.sun.com
```

Ordering Sun Documentation

`fatbrain.com`, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on `fatbrain.com` at:

```
http://www.fatbrain.com/documentation/sun
```

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

```
docfeedback@sun.com
```

Please include the part number of your document, which is on the title page, in the subject line of your email.

CHAPTER 1

Introduction to the System Controller

This chapter introduces you to the system controller. It explains the platform shell and the domain shell. It describes the responsibilities of the platform administrator and the domain administrator . It provides step-by-step procedures and illustrations on how to navigate with the system controller and between the system controller and the Solaris operating environment and the OpenBoot PROM. It also explains how to terminate a system controller session.

The system controller consists of the System Controller board and the system controller software. The system controller provides communication pathways between the system controller and domains.

The system controller software does the following:

- Monitors and controls the system
- Manages hardware
- Configures domains
- Provides the date and time to the Solaris operating environment
- Provides the clock signal used on all system boards
- Provides a platform console and a domain console

The following topics are covered in this chapter:

- "Platform Administration" on page 2
- "Domain Administration" on page 3
- "Accessing the System Controller Main Menu" on page 5
- "System Controller Navigation" on page 10

Platform Administration

The platform administration function provides services for the domain and provides access to hardware available within the platform. With this function, you can manage hardware resources across domains.

Some of the functions include:

- Logically grouping hardware to create domains with the `addboard` and `deleteboard` commands
- Setting passwords with the `password` command
- Monitoring and controlling power to the components within a platform.
- Configuring the system controller, the network, loghosts, and SNMP
- Defining the maximum number of domains a system can support
- Defining access control for CPU/Memory boards and I/O assemblies

Platform Shell

With the platform shell, you can use commands that pertain to the platform. Specifically, the platform shell is the operating environment for the platform administrator, and is the shell where platform tasks can be performed.

You can access the platform shell with a `telnet` session (if your system controller is networked) and/or a serial connection. Log messages go to the platform shell's serial (RS-232) connection (console). The prompt is *schostname*`:SC>`.

Platform Console

A platform console is a platform shell that is connected through the serial (RS-232) port.

The platform console provides messages specific to the platform. However, the platform console does not show the Solaris operating environment messages. These messages are displayed in the domain console.

Domain Administration

Some of the he domain administrator's responsibilities include:

- Controlling the virtual domain keyswitch. A domain administrator can put the keyswitch in on, off, diag, standby, or secure keyswitch positions.
- Managing the domain using the Sun™ Management Center software for the Sun Fire 6800/4810/4800/3800 systems.
- Recovering from errors
- Setting the date, time, and time zone with the `setdate` command. You can set each domain a different setting.
- Configuring domain specific parameter values with the `setupdomain` command.

Domain Shell and Domain Console

This section presents information on these topics:

- "Domain Shell" on page 3
- "Domain Console" on page 4
- "Connecting to a Domain" on page 4
- "Maximum Number of Domains" on page 4

Domain Shell

If . . .	You are connected to . . .
the Solaris operating environment or the OpenBoot PROM is not active in the domain (domain keyswitch is set to standby or off, which means the domain is *inactiv*e)	Domain shell. The prompt is *schostname*:*domainID*>. For example, *schostname*:A>.

Domain Console

If . . .	You can access . . .
the Solaris operating environment is running or the domain is in OpenBoot PROM mode, which means the domain is *active*	Domain console (`ok`, `login`, #, or `%` prompts). Additionally, you can also see POST output, enter the debugger, or enter the OpenBoot PROM

Connecting to a Domain

When you connect to a domain, if the domain is active you will be connected to the domain console. Otherwise, you will be connected to the domain shell.

Maximum Number of Domains

TABLE 1-1 lists the maximum number of domains you can have per system type depending on the partition mode set with the `setupplatform` command.

TABLE 1-1 Maximum Number of Active Domains by System Type and Partition Mode

System Name	Maximum Number of Active Domains in Single Partition Mode	Maximum Number of Active Domains in Dual Partition Mode
Sun Fire 6800 system	2 (A-B)	4 (A-D)*
Sun Fire 4810 system	2 (A,C)	2 (A, C)*
Sun Fire 4800 system	2 (A,C)	2 (A, C)*
Sun Fire 3800 system	2 (A,C)	2 (A, C)*

With the Sun Fire 6800 system, domains A and B are in partition 0 and domains C and D are in partition 1. With the Sun Fire 4810/4800/3800 systems, domain A is in partition 0 and domain C is in partition 1.

For additional information on Repeater boards and domains in both single partition mode and dual partition mode, see “Repeater Board” on page 98.

Accessing the System Controller Main Menu

This section describes the following:

- "Accessing the System Controller Main Menu Using `telnet`" on page 5
- "Accessing the System Controller Main Menu Using the Serial Port" on page 9

Accessing the System Controller Main Menu Using `telnet`

This section describes:

- "To Access the System Controller Main Menu and Enter the Platform Shell Using `telnet`" on page 5
- "To Access the System Controller Main Menu and Enter a Domain Shell Using `telnet`" on page 6

▼ To Access the System Controller Main Menu and Enter the Platform Shell Using `telnet`

1. **Access the system controller main menu by typing** `telnet` *schostname***.**

   where:

   schostname is the system controller host name.

   The system controller main menu is displayed (CODE EXAMPLE 1-1). The following example shows entering the platform shell.

CODE EXAMPLE 1-1 Accessing the System Controller With `telnet` and Entering the Platform Shell

```
% telnet schostname
Trying xxx.xxx.xxx.xxx
Connected to schostname-sc0.
Escape character is '^]'.


System Controller schostname-sc0

    Type 0 for Platform Shell

    Type 1 for domain A
    Type 2 for domain B
    Type 3 for domain C
    Type 4 for domain D

    Input: 0

Platform Shell

schostname-sc0
```

2. **Type `0` to enter the platform shell.**

   You can also type `P` or `p` to enter the platform shell.

   The system controller prompt for the platform or shell you connected to is displayed. CODE EXAMPLE 1-1 shows entering the platform shell whose prompt is *schostname*`:SC>`.

▼ To Access the System Controller Main Menu and Enter a Domain Shell Using `telnet`

1. **Access the system controller main menu by typing `telnet` *schostname*.**

   where:

   schostname is the system controller host name.

   The system controller main menu is displayed (CODE EXAMPLE 1-2).

   The following example shows entering the domain A shell.

CODE EXAMPLE 1-2 Accessing a Domain Shell Using the `telnet` Command

```
% telnet schostname
Trying xxx.xxx.xxx.xxx
Connected to schostname-sc0.
Escape character is '^]'.


System Controller schostname-sc0:

    Type 0 for Platform Shell

    Type 1 for domain A
    Type 2 for domain B
    Type 3 for domain C
    Type 4 for domain D

Input: 1

Connected to Domain A

Domain Shell for Domain A

schostname:A>
```

2. **Enter a domain. Type** `1`**,** `2`**,** `3`**, or** `4` **(or alternatively** `a`**,** `b`**,** `c`**,** `d`**, or** `A`**,** `B`**,** `C`**,** `D`**) to enter the proper domain shell.**

   The system controller prompt for the domain shell you connected to is displayed. CODE EXAMPLE 1-2 shows entering the domain A shell whose prompt is *schostname*`:A>`.

3. **If the domain is active (the domain keyswitch is set to on, diag, or secure which means you are running the Solaris operating environment, are in the OpenBoot PROM, or are running POST), perform the following steps**:

   a. **Press and hold the CTRL key while pressing the] key, to get to the** `telnet>` **prompt.**

b. At the `telnet>` **prompt type** `send break` **(see** CODE EXAMPLE 1-3**).**

CODE EXAMPLE 1-3 Accessing a Domain Shell From the Domain Console

```
ok Ctrl-]
telnet> send break
schostname:A>
```

Accessing the System Controller Main Menu Using the Serial Port

With the serial port, you can connect to one of three types of devices:

- ASCII terminal
- Network terminal server (NTS)
- Workstation

The procedure is different for each type of device.

▼ To Access the System Controller Main Menu if You Connected the System Controller Serial Port to an ASCII Terminal

1. **Connect the system controller serial port to an ASCII terminal.**

   The system controller main menu is displayed.

2. **From the main menu type `0` to enter the platform shell.**

   You can also type `P` or `p` to enter the platform shell (CODE EXAMPLE 1-1).

▼ To Access the System Controller Main Menu if You Connected the Serial Port to a Network Terminal Server (NTS)

1. **Connect the system controller serial port to a Network Terminal Server (NTS).**

2. **Type the following `telnet` command at the Solaris operating environment prompt:**

```
machinename% telnet NTShostname|IPaddress [port]
```

   The system controller main menu is displayed.

3. **Type `0` to enter the platform shell.**

   You can also type `P` or `p`, instead of typing `0`, to enter the platform shell (CODE EXAMPLE 1-1).

▼ To Obtain the Domain Shell Prompt From the Domain Console Prompt

If the domain is active and the domain keyswitch is set to on, diag, or secure (you are running the Solaris operating environment, are in the OpenBoot PROM, or are running POST), perform the following steps:

1. **Press and hold the CTRL key while pressing the] key, to get to the** `telnet>` **prompt.**
2. **At the** `telnet>` **prompt type** `send break`**.**

CODE EXAMPLE 1-4 shows a connection to domain A.

CODE EXAMPLE 1-4 Accessing a Domain Shell From the Domain Console

```
ok Ctrl-]
telnet> send break
schostname:A>
```

System Controller Navigation

This section explains how to navigate between the:

- System controller platform
- System controller domain console
- System controller domain shell

To return to the origin shell, use the `disconnect` command. In a domain shell to connect to the domain console, use the `resume` command. To connect to a domain shell from the platform shell, use the `console` command.

CODE EXAMPLE 1-1 shows how to navigate between the platform shell, the domain shell, the domain console by using the `console` and `disconnect` commands. FIGURE 1-1 also illustrates how to connect to both the domain shell and platform shell from the Solaris operating environment by using the `telnet` command.

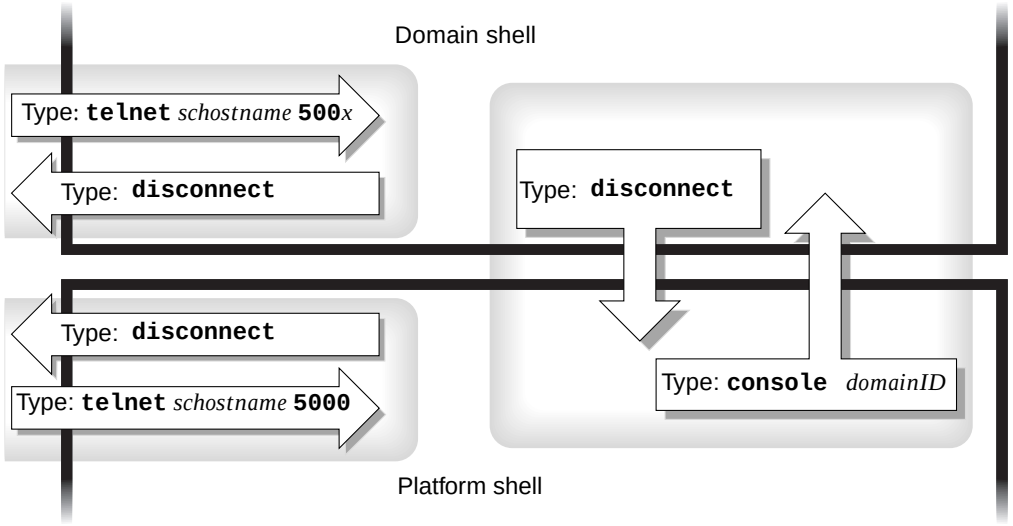

FIGURE 1-1 Navigating Between the Platform Shell and the Domain Shell

where:

In the `telnet` command in FIGURE 1-1, `5000` is the platform shell.

x is:

- 1 for domain A
- 2 for domain B
- 3 for domain C
- 4 for domain D

In the `console` command, *domainID* is a, b, c, or d.

Note – By typing `telnet` *schostname* `500`*x*, you *directly* enter the platform shell or one of the domain shells. You will bypass the system controller main menu.

FIGURE 1-2 illustrates how to navigate between the Solaris operating environment, the OpenBoot PROM, and the domain shell. FIGURE 1-2 assumes that the Solaris operating environment is running.

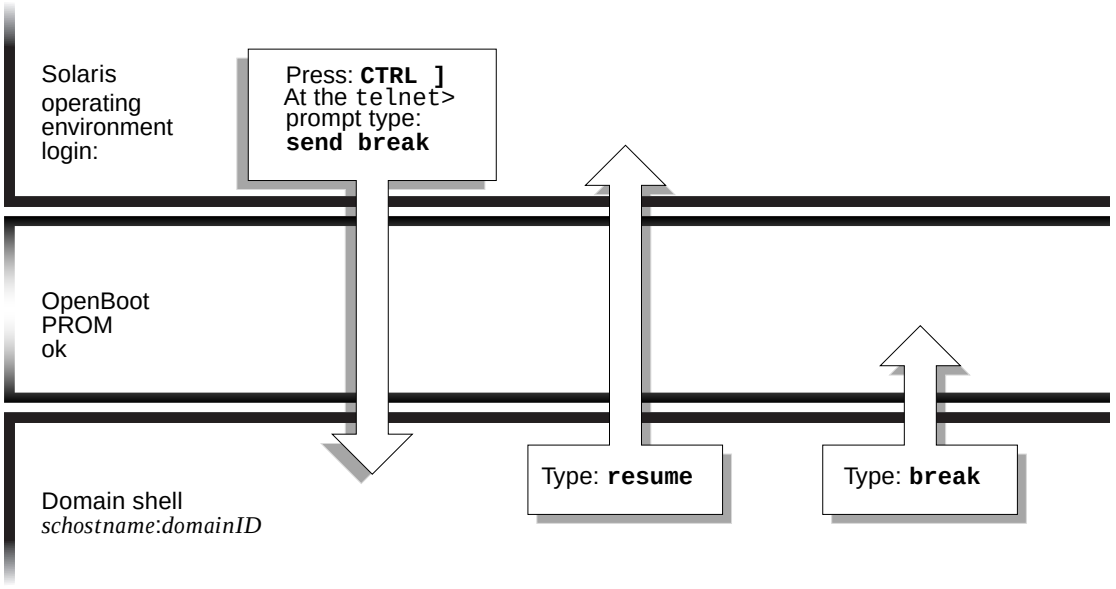

FIGURE 1-2 Navigating Between the Domain Shell, the OpenBoot PROM, and the Solaris Operating Environment

---

Note – Note that in FIGURE 1-2 typing the `break` command suspends the Solaris operating environment.

FIGURE 1-3 illustrates how to navigate between the OpenBoot PROM and the domain shell. This figure assumes that the Solaris operating environment is *not* running.

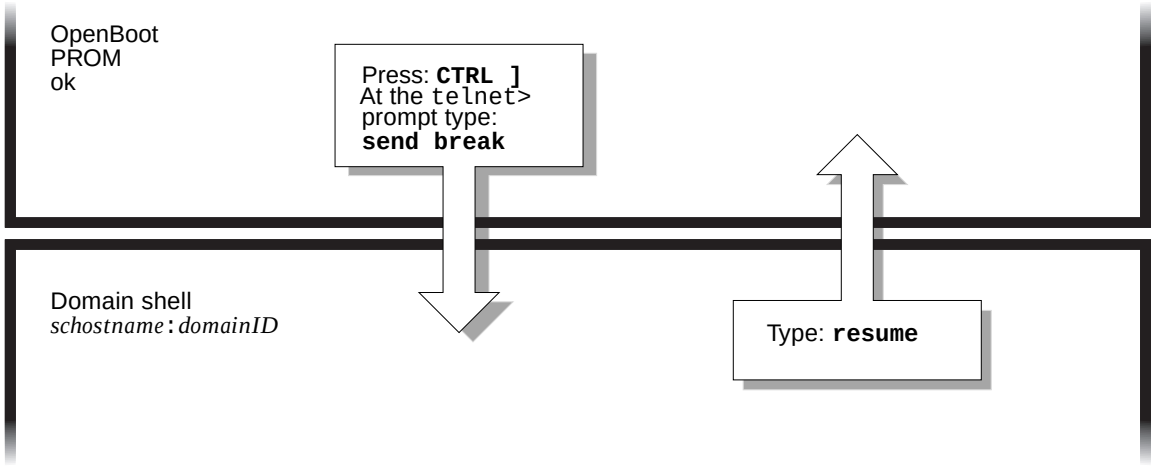

FIGURE 1-3 Navigating Between the OpenBoot PROM and the Domain Shell

When you enter a domain console, you will be connected with the Solaris operating environment console. If either POST or the OpenBoot PROM is running, you will be connected with either the POST or the OpenBoot PROM output.

▼ To Enter the Domain Console From the Domain Shell If the Domain Is Inactive

- **Type** `setkeyswitch on` **in the domain shell.**

```
schostname:A> setkeyswitch on
```

The domain console is only available when the domain is active. To make the domain active, you must turn the keyswitch on. You will be automatically switched from the domain shell to the domain console.

This action powers on and initializes the domain. The domain will go through POST and then the OpenBoot PROM. If the OpenBoot PROM `auto-boot?` parameter is set to true, the Solaris operating environment will boot.

▼ To Enter the Domain Shell From the Domain Console

1. **Press and hold the CTRL key while pressing the] key to get to the** `telnet>` **prompt (**CODE EXAMPLE 1-5**).**
2. **Type** `send break` **at the** `telnet` **prompt.**

   The *schostname:*A> (or B>, C>, D>) prompt is displayed.

CODE EXAMPLE 1-5 Obtaining a Domain Shell From the Domain Console

```
ok Ctrl-]
telnet> send break
schostname:A>
```

▼ To Get Back to the Domain Console From the Domain Shell

1. **Type** `resume`**::**

```
schostname:A> resume
```

2. **Press the Enter key to get a prompt.**

   If the domain is not active, (the Solaris operating environment or the OpenBoot PROM is *not* running), the system controller stays in the domain shell.

```
Domain A is not active.
schostname:A>
```

▼ To Enter a Domain From the Platform Shell

Note – This example shows entering an inactive domain.

- **Type**:

```
schostname:SC> console -d a

Connected to Domain A

Domain Shell for Domain A

schostname:A>
```

If the OpenBoot PROM is running, you are returned to the domain A console. If the keyswitch is set to `off` or `standby`, you are returned to the domain A shell.

Note – To enter another domain, type the proper *domainID* `b`, `c`, or `d`, instead of typing `a`.

▼ To Terminate a Session With `telnet` If You Are Connected to the Ethernet Port

- **Type the `disconnect` command at the domain shell prompt.**

```
schostname:A> disconnect
Connection closed by foreign host.
machine_name_prompt%
```

Your system controller session terminates.

▼ To Terminate a Session With `tip` If You Are Connected to the Serial Port

If you are connected to the System Controller board with the serial port, use the `disconnect` command to terminate the system controller session then use a `tip` (Transfer Internet Protocol) command to terminate your `tip` session.

1. **Type `disconnect` to disconnect from the system controller session.**
2. **At the Solaris operating environment prompt, type `~.` to terminate your `tip` session.**

```
schostname:A> disconnect
Connection closed by foreign host.
machine_name_prompt% ~.
```

Your system controller session terminates.

CHAPTER 2

System Controller Alphabetical Command Reference

This chapter describes system controller command line interface, command syntax, and command arguments. It provides, in tabular form, a summary of all of the system controller commands. It also provides complete descriptions and examples for each system controller command.

System Controller Syntax, Command Names, Command Arguments, and Comments

Command Syntax

The general syntax of system controller commands is:

command_name [option flags] [*arguments*]

Note – The system controller commands can be partially typed if what is typed is enough to make the command unique.

Command Names

Most system controller commands are in the form of *verbnoun*. Command names are in the English language. System controller commands are case insensitive.

Command Arguments

Some system controller commands have arguments and some system controller commands do not have arguments. Depending on the command, some arguments are followed by flags and others are not. A few examples include:

- `help`
- `help -v`
- `showboards sb0`
- `showboards -v sb0`

Device Names

TABLE 2-1 shows the device names that can be managed in the platform shell. You can type the device names in either upper case or lower case in both TABLE 2-1 and TABLE 2-5.

Note – The devices in TABLE 2-1 and TABLE 2-2 depend on the system you have. For example, only the Sun Fire 6800 system can have six power supplies and six CPU/Memory boards. In a domain, you can manage devices only if they are assigned to a domain.

TABLE 2-1 Devices That Can Be Managed in the Platform Shell

Device Description	Device Name
Power grids	`GRID0, GRID1`
Power supplies	`PS0, PS1, PS2, PS3, PS4, PS5`
CPU/Memory boards	`SB0, SB1, SB2, SB3, SB4, SB5`
I/O assemblies	`IB6, IB7, IB8, IB9`
Repeater boards	`RP0, RP1, RP2, RP3`
Fan trays	`FT0, FT1, FT2, FT3`
System controller	`SC0, SC1`

TABLE 2-2 shows the devices that can be managed in a domain.

TABLE 2-2 Devices That Can Be Managed in a Domain

Device Description	Device Name
CPU/Memory boards	`SB0, SB1, SB2, SB3, SB4, SB5`
I/O Assemblies	`IB6, IB7, IB8, IB9`

Board States

TABLE 2-3 lists the board states. These board states are displayed by the `showboards` command. For some commands, such as `addboard` and `deleteboard`, you need to determine the board state with `showboards` before you can execute the `addboard` or `deleteboard` command.

TABLE 2-3 Board States

Board States	Description
Available	The board is not assigned to any domain.
Assigned	The board belongs to a domain, but the hardware has not been configured or it is not in use.
Active	The board is being actively used by the domain to which it has been assigned. You cannot reassign an active board.

To determine the board state, use the `showboards` command and look under the `State` header.

Status

TABLE 2-4 describes the Status field displayed in the `showboards` command.

TABLE 2-4 Status Field Displayed in the `showboards` Command

Test Status	Description
Passed	All board components passed testing.
Failed	The board failed testing and is not usable. This could also indicate corrupt or incompatible firmware.
OK	The component is functioning properly (power supply, fan tray, or Repeater board).
Under Test	The domain is running POST (power-on self-test).
Not Tested	The board is not been tested.
Degraded	Certain components on the board have failed or are disabled. A board is in degraded mode when there are still usable parts on the board.
-	The slot is empty or not applicable for this device.

Comments

You can separate system controller commands with a semicolon (;) on the same line. For example:

```
schostname:SC> addboard -d a sb2;deleteboard sb3
```

In the previous example, both the `addboard` and `deleteboard` commands are executed.

A pound sign (#) signifies the start of a comment on the current line. For example:

```
schostname:SC> addboard -d a sb2 #this text is ignored
```

In the previous example, the `addboard` command is executed. Everything you type after the # and before pressing the Return key is ignored.

If you type the following line, board `sb2` is added to domain A and the `deleteboard` command is not executed.

```
schostname:SC> addboard -d a sb2 # deleteboard sb2
```

Another example of the pound sign (#) signifying the start of a comment at the beginning of a command line is:

```
schostname:SC> # this text is ignored
```

Clearing an Entry

You use the dash (-) to clear the entry in a command.

System Controller Command Summary

TABLE 2-5 lists and describes the system controller commands and how you can access them. Many commands are accessible from both the platform shell and the domain shell. The system controller commands may differ in how they are used, the effect of the command, and the scope of the command between the platform and domain shells.

TABLE 2-5 System Controller Command Summary

Command	Description	Platform Shell	Domain Shell
`addboard`	Assigns a board to a domain.	x	x
`break`	Sends a Break signal to the domain.		x
`connections`	Displays connections to the system controller or to a domain.	x	x
`console`	Connects to a domain from the platform shell.	x	
`deleteboard`	Unassigns a board or a slot from a domain.	x	x
`disablecomponent`	Adds a component to the blacklist.	x	x
`disconnect`	Disconnects the current specified connection.	x	x

TABLE 2-5 System Controller Command Summary *(Continued)*

Command	Description	Platform Shell	Domain Shell
`dumpconfig`	Saves the system controller configuration to a server.	x	
`enablecomponent`	Deletes a component from the blacklist.	x	x
`flashupdate`	Updates the flash PROMs on CPU/Memory boards, I/O assemblies, and the System Controller boards.	x	
`help`	Provides basic help information for system controller commands.	x	x
`history`	Shows the command history with date and time stamps.	x	x
`password`	Sets the shell password. There are separate passwords for the platform connection and each domain connection.	x	x
`poweroff`	Powers off components.	x	x
`poweron`	Powers on components.	x	x
`reboot`	Reboots the system controller.	x	
`reset`	Resets the domain in the domain shell or the other system controller in the platform shell.	x	x
`restoreconfig`	Restores the system controller configuration from a server.	x	
`resume`	Exits the domain shell and resumes access to the domain console.		x
`setdate`	Sets the time, date, and time zone for the platform and each domain.	x	x
`setdefaults`	Resets the system controller to the default values.	x	x
`setkeyswitch`	Changes the keyswitch position.	x	x
`setupdomain`	Configures domain specific variables.		x
`setupplatform`	Configures platform specific variables.	x	
`showboards`	Shows status, assignment, and board information for boards in the system.	x	x
`showcomponent`	Displays detailed information about a component.	x	x
`showdate`	Displays the time, date, and the timezone for the platform or for domains.	x	x

TABLE 2-5 System Controller Command Summary *(Continued)*

Command	Description	Platform Shell	Domain Shell
`showdomain`	Displays the configuration and status of the domain.		x
`showenvironment`	Displays the current environmental status, temperatures, currents, and voltages.	x	x
`showkeyswitch`	Displays the virtual keyswitch setting.	x	x
`showlogs`	Displays the logs.	x	x
`showplatform`	Displays the configuration of the platform, the status of a domain, and SNMP information.	x	
`showsc`	Displays the system controller uptime and version.	x	
`testboard`	Tests the CPU / Memory board you specify at the command line.	x	x

Status

TABLE 2-6 describes the Status field displayed in the `showboards` command.

TABLE 2-6 Status Field Displayed in the `showboards` Command

Test Status	Description
Passed	All board components passed tests.
Failed	A component fails when there are no usable parts on the board.
OK	The component is functioning properly (power supply, fan tray, or Repeater board).
Under Test	The domain is running POST (power-on self-test).
Not Tested	No testing has been done.
`– (dash)`	The slot is empty or disabled. Not applicable for this device.
Degraded	Certain components on the board have failed or are in the blacklist. A board is in degraded mode when there are still usable parts on the board.

Domain States

In the `showplatform` and `showdomain` commands, one of the fields in the command output is domain status. The major values of domain status are:

- Active-Solaris
- Active-OpenBoot PROM
- Active-Booting
- Running POST
- Standby
- Powered off

Alphabetical Listing of System Controller Commands

The following sections describe the system controller commands.

addboard

Assigns a board or a slot to a domain.

Scope

platform shell, domain shell

Syntax

`addboard -d` *domainID* *system_board_name* [*system_board_name . . .*]

`addboard -h`

Options / Parameters

`-h` displays help for this command.

system_board_name is the board to be added. Possible values for *system_board_name* are `sb0`–`sb5` (CPU / Memory board) or `ib6` - `ib9` (I / O assembly).

`-d` *domainID* is the domain where the board will be added. The *domainID* is a, b, c, or d.

Description

Assigns *system_board_name* to the specified domain within the platform shell or to the current domain. The board state must be *Available*. To display the board states, use the `showboards` command. For more information on board states, see “Board States” on page 19.

In order to assign a board to a domain, the board name must be listed in the Access Control List (ACL) for domain shells. When a board is listed in the ACL, the system controller software is allowed to process `addboard` requests on that board.

The Access Control List (ACL) is ignored in the platform shell. The platform shell always has permission to add or delete boards. Even if a board is not in the domain’s ACL, when you type the `addboard` command from the platform shell, this overrides the ACL.

If a board is not present, the command assigns ownership of the slot to the specified domain.

See Also

`deleteboard`, `setupplatform`, `showboards`, `showdomain`, `showplatform`, and *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for a step-by-step procedure on assigning boards to a domain).

Example—Platform Shell

To add board name `sb2` to domain A, type:

CODE EXAMPLE 2-1 `addboard` Example in the Platform Shell

```
schostname:SC> addboard -d a sb2
```

Although a message is sent to the Solaris operating environment about a new board being added, there is not a client in the Solaris operating environment that reads these messages. Therefore, these messages will not be seem by the Solaris operating environment.

Example—Domain Shell

To add board name `sb2` to the current domain, type:

CODE EXAMPLE 2-2 `addboard` Example in the Domain Shell

```
schostname:A> addboard sb2
```

If you cannot assign a board to a domain in the domain shell, this is most likely because the board is not listed in the ACL. See the “Description“ section for more information. Another possibility is that the domain is currently in use. An error message will tell you the reason why you can not perform the operation.

break

Sends a break signal to the domain console.

Scope

domain shell

Syntax

```
break [-y|-n]
break -h
```

Options / Parameters

`-h` displays help for this command.

`-y` answers yes to the warning message. Does not prompt for confirmation.

`-n` answers no to the warning message. Does not execute this command if confirmation is requested.

Description

Sends a Break signal to the domain console.

Note – This command pauses the Solaris operating environment.

When the Solaris operating environment is running in the domain, the usual effect of the `break` signal is to force entry into OpenBoot PROM or the debugger. The Solaris operating environment will ignore the break signal if the keyswitch is set to secure.

See Also

`setkeyswitch,showkeyswitch`

Example

CODE EXAMPLE 2-3 shows using the `break` command to pause the Solaris operating environment and enter the OpenBoot PROM.

CODE EXAMPLE 2-3 `break` Command Example in the Domain Shell

```
schostname:A> break
ok
```

connections

Displays connections to the system controller or a domain.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`connections [-d` *domainID*`]`

`connections -h`

Syntax for the Domain Shell

`connections [-h]`

Options / Parameters

`-h` displays help for this command.

`-d` *domainID* shows connections to the specified domain (a, b, c, or d).

Description

Displays hosts currently connected to the system controller. The platform shell shows the connections to the platform shell and to each domain shell (or to the specified domain). The domain shell displays the information shown in CODE EXAMPLE 2-4, which is described in TABLE 2-7.

See Also

`disconnect`

Example—Platform Shell

CODE EXAMPLE 2-4 `connections` Command for the Platform Shell

```
schostname:SC> connections
ID      Hostname          Idle      Connected on   Connected to
--      --------          ----      ------------   ------------
1       XXXXXXX           -         Jan 01 00:00   Platform
2       XXX               -         Jan 01 00:21   A
```

TABLE 2-7 describes the headers in the output for CODE EXAMPLE 2-4.

TABLE 2-7 Header Description for CODE EXAMPLE 2-4

Header Column in CODE EXAMPLE 2-4	Description
ID	Connection ID. This is a unique number. Use this number to disconnect a specific session.
Hostname	Source of the connection for this session. If the value is `localhost`, the connection was established through the platform.
Idle	Amount of time the session is sitting without any activity. Displays a dash (-) if the session is active.
Connected on	Date and time the connection was initiated. The format is `Mon dd hh:mm`.
Connected to	Shell or console this session is connected to. Possible values are `Platform` or *domainIDs* a, b, c, or d.

console

Connects to a domain from the platform.

Scope

platform shell

Syntax

`console [-d]` *domainID*

`console -h`

Options / Parameters

`-h` displays help for this command.

`[-d]` *domainID* is the domain to connect to and is `a`, `b`, `c`, or `d`. The `-d` parameter is optional and does not need to precede the *domainID*.

Description

Opens a domain connection. You use the `console` command to navigate from the platform to a domain. If a password was created for connecting to this domain, you must enter the password.

See Also

`disconnect`

Examples—Platform Shell

When there is no password initialized for this domain:

CODE EXAMPLE 2-5 `console` Example With the `-d` Option and No Password Assigned Accessing Domain B

```
schostname:SC> console -d b

Connected to Domain B

Domain Shell for Domain B

schostname:B>
```

When there is no password initialized for this domain, you could also type:

CODE EXAMPLE 2-6 `console` Example Without the `-d` Option and No Password Assigned Accessing Domain B

```
schostname:SC> console b

Connected to Domain B

Domain Shell for Domain B

schostname:B>
```

When a password is set for this domain:

CODE EXAMPLE 2-7 `console` Example With the `-d` Option and A Password Assigned Accessing Domain A

```
schostname:SC> console -d a

Password:

Permission Granted
Connected to Domain A

Domain Shell for Domain A

schostname:A>
```

When a password is set for this domain, you could also type:

CODE EXAMPLE 2-8 `console` Example Without the `-d` Option and A Password Assigned Accessing Domain A

```
schostname:SC> console a

Password:

Permission Granted
Connected to Domain A

schostname:A>
```

deleteboard

Unassigns a board or a slot from the domain where it is currently assigned.

Scope

platform shell, domain shell

Syntax for the Platform Shell and the Domain Shell

`deleteboard` *system_board_name* [*system_board_name* `. . .`]

`deleteboard [-h]`

Options / Parameters

`-h` displays help for this command.

system_board_name is the board to be deleted. Values are `sb0` to `sb5` (CPU / Memory board) and `ib6` to `ib9` (I / O assembly).

Description

Unassigns a board from the domain where it is currently assigned. The CPU / Memory board or the I / O assembly board status must be in the *Assigned* state. The board name must be listed in the Access Control List (ACL) for domain shell from which you want to unassign the board.

Type the `showboards` command to display the board state. A slot does not need to be occupied with a board before you can unassign it from a domain.

See Also

`addboard`, `showboards`, `setupplatform`, `showdomain`, `showplatform`, and the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for step-by-step procedure on unassigning boards from a domain).

Example—Platform Shell

To delete I/O assembly `ib7`, type:

CODE EXAMPLE 2-9 `deleteboard` Example Showing Deleting I/O Assembly 7

```
schostname:SC> deleteboard ib7
```

The platform shell always has permission to add or delete boards.

Example—Domain Shell

To delete CPU/Memory board `sb3` from the current domain, domain A, type:

CODE EXAMPLE 2-10 `deleteboard` Example Showing Deleting CPU/Memory Board 3

```
schostname:A> deleteboard sb3
```

The domain shell can delete boards only from the current domain.

disablecomponent

Adds a component to the blacklist.

Scope

platform shell, domain shell

Syntax

`disablecomponent` *component_name* [*component_name* . . .]

`disablecomponent -h`

Options / Parameters

`-h` displays help for this command.

component_name is (see TABLE 2-8 and TABLE 2-9):

- *system_board_name* / *port* / *physical_bank* / *logical_bank* for the CPU / Memory board
- *system_ board_name* / *port* / *bus* for an I / O assembly
- *system_board_name* / *card* for an I / O assembly

Note – You can use partial component names.

TABLE 2-8 *component_name* Descriptions for the CPU/Memory Board Used in the `disablecomponent` Command

Board or Device	Component Name
CPU system	*board_name*/*port*/*physical_bank*/*logical_bank*
CPU/Memory boards (*system_board_name*)	SB0, SB1, SB2, SB3, SB4, SB5
Ports on the CPU/Memory board	P0, P1, P2, P3
Physical memory banks on CPU/Memory boards	B0, B1
Logical banks on CPU/Memory boards	L0, L1, L2, L3

TABLE 2-9 *component_name* Descriptions for the I/O Assembly Used in the `disablecomponent` Command

Board or Device	Component Name
I/O assembly system	*board_name*/*port*/*bus* or *board_name*/*card*
I/O assemblies (*system_board_name*)	IB6, IB7, IB8, IB9
Ports on the I/O assembly	P0, P1
Buses on the I/O assembly	B0, B1
I/O cards in the I/O assembly	C0, C1, C2, C3, C4, C5, C6, C7—the number of cards varies with the I/O assembly

Description

This command adds a component to the blacklist. Blacklisting provides a list of system board components that will not be tested and will not be configured into the domain when you change the domain's keyswitch from an inactive state to an active state or when the domain is rebooted.

The blacklists are stored in non-volatile memory. Blacklist a component or device if you believe it may be failing intermittently.

Devices can be the following (see TABLE 2-8 and TABLE 2-9):

- CPU ports (CPU processors)
- Physical and logical memory banks
- I/O ports
- I/O buses
- I/O cards

The platform shell blacklists supersede the domain shell blacklists. For example, if a component is disabled in the platform shell, it will always be disabled in all domains.

Blacklisting from the platform shell applies to all domains. Blacklisting in a domain applies only to the current domain. If you disable a component in a domain and then move the component to another domain, the component is not disabled.

See Also

`enablecomponent`, `showcomponent`, and the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for step-by-step procedure on adding a component to the blacklist).

Examples

This example of the `disablecomponent` command adds *system_board_name* `sb4` to the blacklist.

CODE EXAMPLE 2-11 `disablecomponent` Command Example Adding `sb4` to the Blacklists

```
schostname:A> disablecomponent sb4
```

This example of the `disablecomponent` command adds *system_board_name* `sb0` and CPU port 3 to the blacklist. Note that any memory banks on this CPU processor port are unreachable and are implicitly disabled.

CODE EXAMPLE 2-12 `disablecomponent` Command Example Adding `sb0` CPU Port 3 to the Blacklist

```
schostname:A> disablecomponent sb0/p3
```

This example of the `disablecomponent` command adds *system_board_name* `sb2`, CPU processor port 3, physical bank 0 to the blacklists. Note that any logical banks belonging to this physical bank are disabled.

CODE EXAMPLE 2-13 `disablecomponent` Command Example Adding `sb2`, CPU Port 3, and Physical Bank 0 to the Blacklists

```
schostname:A> disablecomponent sb2/p3/b0
```

This example of the `disablecomponent` command adds *system_board_name* `ib9`, port 0, bus 1 and *board_name* `ib8`, I/O card 2 to the blacklists. Disabling I/O card 2 shuts down the power to that I/O card.

CODE EXAMPLE 2-14 `disablecomponent` Command Example Adding I/O Assembly 9, Port 0 and Bus 1 and Also I/O Assembly 8, I/O Card 2 to the Blacklists

```
schostname:A> disablecomponent ib9/p0/b1 ib8/c2
```

disconnect

Disconnects the current or specified connection.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
disconnect [ID]
```

```
disconnect -h
```

Syntax for the Domain Shell

```
disconnect [-h]
```

Options / Parameters

`-h` displays help for this command.

ID (platform shell only) is the connection to be terminated. *ID* is the number displayed under the heading ID for the `connections` command.

Description

Terminates a connection to the system controller. If this command is used with no arguments, it disconnects the current session. For illustrations of the disconnect command, see FIGURE 1-1 in “System Controller Navigation” on page 10.

If the connection was initiated from another system, you will be returned to the remote host. If the session was initiated from the platform shell, this command displays the platform shell prompt. If you are connected from the serial port, then you will return to the main menu.

See Also

`connections`

Examples

To disconnect a session from the system controller platform shell with a `telnet` connection (CODE EXAMPLE 2-15), type:

CODE EXAMPLE 2-15 Disconnecting a Session From the System Controller Platform Shell

```
schostname:SC> disconnect
Connection closed by foreign host.
```

If you are connected to the system controller platform console with the serial connection, you will see the following after typing `disconnect`:

CODE EXAMPLE 2-16 Disconnecting a Session From the System Controller Platform Console

```
schostname:SC> disconnect

    Type 0 for Platform Shell

    Type 1 for domain A
    Type 2 for domain B
    Type 3 for domain C
    Type 4 for domain D

    Input:
```

To disconnect a session from a system controller domain shell, type:

CODE EXAMPLE 2-17 `disconnect` Example Showing Disconnecting a Session From the Domain A Shell

```
schostname:A> disconnect
Connection closed by foreign host.
```

Note – When you disconnect from a domain shell, you will obtain the parent shell prompt (either the remote host prompt or the platform shell prompt).

---

dumpconfig

Saves the platform and domain configurations to a server.

Scope

platform shell

Syntax

`dumpconfig -f` *url*

`dumpconfig -h`

Options / Parameters

`-h` displays help for this command.

`-f` specifies the URL, which can only be `ftp`. For example:

`ftp://`*userid*`:`*password*`@`*host*`/`*path*

`ftp://`*host*`/`*path*

Note – The hostname you enter must be a complete host name (not a partial one).

Note – The directory specified in the URL path must have write permission, including user access. Also, make sure that the data files are present in the URL.

Description

Use this command after you complete the initial configuration of the platform and the domains. This command saves the system controller configuration to a server. The data files include *schostname*`.nvci` and *schostname*`.tod`.

Caution – Run this command when you change platform or domain configurations. For more information, see the following bulleted list of system controller commands that change the configuration. Also, run this command whenever you change the hardware configuration.

Invoking this command is *very important* because if the System Controller board fails and you did not use this command to save the platform and domain configurations, you will have to manually reconfigure the platform and the domains.

Run the `dumpconfig` command again after configuration changes are made to the system controller, for example with the following system controller commands:

- `setupdomain`
- `setupplatform`
- `setdate`
- `addboard`
- `deleteboard`
- `enablecomponent`
- `disablecomponent`
- `password`
- `setkeyswitch`

or whenever you change the hardware configuration. By running again this command, the new platform and domain configurations are saved to a server.

This command is intended to be used as part of the procedure in replacing a failed System Controller board. You *should not* use this command to revert to an old configuration. The `showplatform`, `showdomain`, and `showboards` system controller commands display all of the configuration and can be used as a reference.

If you need to replace the System Controller board, this configuration information can be used to restore the platform and domain configurations to the replacement System Controller board.

See Also

`restoreconfig`

Example

TABLE 2-10 `dumpconfig` Example

schostname:SC> **dumpconfig -f ftp://***hostname***/***path_name* Created: ftp://*host*/*path_name*/*schostname*.nvci Created: ftp://*host*/*path_name*/*schostname*.tod

enablecomponent

Deletes a component from the blacklist.

Scope

platform shell, domain shell

Syntax

`enablecomponent` *component_name* [*component_name . . .*]

`enablecomponent -h`

Options / Parameters

`-h` displays help for this command.

component_name is (TABLE 2-11 and TABLE 2-12):

- *system_board_name* / *port* / *physical_bank* / *logical_bank* for CPU / Memory boards
- *system_board_name* / *port* / *bus* for an I / O assembly
- *system_board_name* / *card* for an I / O assembly

TABLE 2-11 *component_name* Descriptions for the CPU / Memory Board Used in the `enablecomponent` Command

Board or Device	Component Name
CPU system	*board_name* / *port* / *physical_bank* / *logical_bank*
CPU / Memory boards (*system_board_name*)	SB0, SB1, SB2, SB3, SB4, SB5
Ports on the CPU / Memory board	P0, P1 ,P2, P3
Physical memory banks on CPU / Memory boards	B0, B1
Logical banks on CPU / Memory boards	L0, L1, L2, L3

TABLE 2-12 *component_name* Descriptions for the I/O Assembly Used in the `enablecomponent` Command

Board or Device	Component Name
I/O assembly system	*board_name*/*port*/*bus* or *board_name*/*card*
I/O assemblies (*system_board_name*)	IB6, IB7, IB8, IB9
Ports on the I/O assembly	P0, P1
Buses on the I/O assembly	B0, B1
I/O cards in the I/O assembly	C0, C1, C2, C3, C4, C5, C6, C7—the number of cards varies with the I/O assembly

Description

Removes a component from the blacklist. Blacklisting provides a list of system board components that will not be tested and will not be configured into the domain when you change the keyswitch setting from an inactive state to an active state or when the domain is rebooted. The blacklists are stored in non-volatile memory.

Components can be the following:

- CPU ports (CPU processors)
- Physical and logical memory banks
- I/O ports
- I/O buses
- I/O cards

Using this command, the platform shell blacklists supersede the domain shell blacklists. For example, if a component is enabled in the platform shell, it will be enabled in all domains.

To enable a component that you previously disabled with the `disablecomponent` command in one or more shells or the platform, you must enable it in the same shells it was disabled.

See Also

`disablecomponent`, `showcomponent`, and the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for step-by-step procedure on removing a component from the blacklist).

Examples

This example of the `enablecomponent` command enables CPU/Memory board `sb4`. This removes CPU/Memory board `sb4` from the blacklist.

TABLE 2-13 `enablecomponent` Command Example Enabling CPU/Memory Board 4

```
schostname:A> enablecomponent sb4
```

This example of the `enablecomponent` command enables I/O assembly 6, port 1. This removes port 1 of I/O assembly 6 from the blacklist.

CODE EXAMPLE 2-18 `enablecomponent` Command Example Enabling I/O Assembly 6 and Port 1

```
schostname:A> enablecomponent ib6/p1
```

flashupdate

Updates the flash PROMs in the system controller and the system boards (CPU/Memory boards and I/O assemblies). The source flash image can be on a server or another board of the same type.

Scope

platform shell

Syntax

`flashupdate [-y|-n] -f` *url* `all|systemboards|scapp|rtos|`*domainID*

`flashupdate [-y|-n] -f` *url board* [*board . . .*]

`flashupdate [-y|-n] -u`

`flashupdate [-y|-n] -c` *source_board destination_board* [*destination_board . . .*]

`flashupdate -h`

Options/Parameters

`-h` displays help for this command.

`-y` does not prompt for confirmation.

`-n` does not execute this command if confirmation is required.

`-f` specifies a URL as the source of the flash images

url is the URL to the directory containing the flash images.

where:

`ftp://`*hostname/path*

`ftp://[`*userid*`:`*password*`@]`*hostname/path*
(userid:*password@* is dependent on the system setup)

`http://`*hostname/path*

`http://[`*userid*`:`*password*`@]`*hostname/path*
The hostname you enter must be a complete host name (not a partial one).

board is the board name.

`-c` specifies a board as the source of flash images.

- *source_board* is the source board for the flash images.
- *destination_board* is the destination board for the flash images.

`-u` automatically updates all of the boards from the board with the highest revision.

`all` updates the system controller and all system boards (CPU/Memory boards and I/O assemblies). This option does not update the system controller real time operating system (`rtos`).

`system_boards` are all CPU/Memory boards and I/O assemblies.

`scapp` updates the current system controller. Updating the system controller reboots the system controller twice. To update the other system controller, you must run the `flashupdate` command from the other system controller. Use `showsc` to determine which system controller is active.

`rtos` updates the real time operating system for the system controller. This action reboots the system controller twice.

domainID (a, b, c or d) is the domain where the flash PROMs on the board will be updated.

Caution – Any boards without power will not be updated.

Description

Updates the flash PROMs on the System Controller board, CPU/Memory board, I/O assembly, and the port of the system controller. There are no flash PROMs on Repeater boards.

When you update the flash images, if you can do so, make sure the images are compatible. To verify the compatibility of the versions of `scapp`, CPU/Memory board, and I/O assembly PROMs, use the `showboards -p proms` command. There is currently no method to determine that the `scapp` and `rtos` versions are compatible. Check all versions before you upgrade with the `flashupdate` command. If you fail to check all versions, the domains may crash or the system controller would fail to reboot normally.

All boards that are specified must be powered on and must be capable of being updating with the `flashupdate` command. Otherwise, the `flashupdate` command will not execute. Boards that are in a secure domain will not be updated.

Any time boards, the `scapp`, or `rtos` images are upgraded, check the Release Notes for any notes or special procedures.

Caution – Under normal circumstances, the `flashupdate` command will complete successfully. However, if the flashupdate command is terminated abnormally (such as a power failure, a failed network connection, and so on), the system controller goes into single user mode.

CODE EXAMPLE 2-19 is a sample output showing the system controller entering single-user mode.

CODE EXAMPLE 2-19 Example of the `flashupdate` Command Entering Single-User Mode

```
RTOS version: 17
ScApp version: 5.11.3
SC POST diag level: min

Auto Flashupdate
S_errno_ECONNREFUSED
http://hostname/5.11.1/sgrtos.flash: cannot open file

Retrieving: http:/hostname/5.11.1/sgrtos.flash
Flashupdate failed.

Single User Mode

The date is Thursday, February 15, 2001, 9:22:00 AM PST.

Feb 15 09:22:03 schostname Chassis-Port.SC: WARNING: hostid
information is missing
Feb 15 09:22:03 schostname Chassis-Port.SC: Clock source: 75MHz
Feb 15 09:22:05 schostname Chassis-Port.SC: Starting Maintenance
Thread

Platform Shell

schostname:SC>
schostname:SC> help

Platform Shell commands:

dumpconfig      -- save the system controller configuration to a
server
flashupdate      -- update flash prom images
help             -- show help for a command or list commands
history          -- show shell command history
reboot           -- reboot the system controller
reset            -- reset the other system controller
restoreconfig    -- restore the system controller configuration
from a server

schostname:SC>
```

To bring the system controller out of single-user mode and back into its normal operating mode, type the `flashupdate` command with a valid URL or use the `reboot` command to boot the system controller with an older firmware version.

See Also

Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual (for step-by-step procedures on how to update the firmware using `flashupdate`).

`flashupdate` Command Platform Shell—Examples

Note – In the following examples, since the output is very long, only the command syntax you type is in the code box.

Note – For the `flashupdate all`, `flashupdate all rtos`, `flashupdate scapp`, and the `flashupdate rtos` commands, all of these commands cause the system controller to reboot twice—one time to upgrade and one time to boot the new firmware.

- To update the flash PROMs on the system controller and all CPU/Memory boards and I/O assemblies, make sure that all boards and assemblies are powered on. If a board or assembly is powered off, the flashupdate operation will fail. Type:

```
schostname:SC> flashupdate -f ftp://host/path all
```

  That command reboots the system controller.

- To update the flash PROMs on all CPU/Memory boards, I/O assemblies, and the system controller real time operating system (`rtos`), type:

```
schostname:SC> flashupdate -f ftp://host/path all rtos
```

  That command reboots the system controller.

- To update the flash PROMs on the system controller, perform this procedure from the platform console or watch the console output. Type:

```
schostname:SC> flashupdate -f ftp://host/path scapp
```

That command reboots the system controller.

- To update the active System Controller board and the system controller real time operating system (`rtos`), perform this procedure from the platform console or watch the console output. Type:

CODE EXAMPLE 2-20 `flashupdate` Example Updating the Active System Controller Board and the System Controller Real Time Operating System

```
schostname:SC> flashupdate -f ftp://host/path scapp rtos
```

This reboots the system controller.

- To upgrade the flash PROMs on all CPU/Memory boards and I/O assemblies (system boards), make sure that all boards and assemblies are powered on. If a board or assembly is powered off, the flashupdate operation will fail. Type:

CODE EXAMPLE 2-21 `flashupdate` Example Updating System Boards (CPU/Memory Boards and I/O Assemblies)

```
schostname:SC> flashupdate -f ftp://host/path systemboards
```

Reboot the domain containing the system boards that were updated in order to take advantage of the new features and bug fixes.

- To upgrade the flash PROMs on CPU/Memory board 4 (SB4), type:

CODE EXAMPLE 2-22 `flashupdate` Example Updating CPU/Memory Board 4

```
schostname:SC> flashupdate -f ftp://host/path sb4
```

Reboot the domain containing the CPU/Memory board that was updated in order to take advantage of the new features and bug fixes.

- To upgrade the flash PROMs on I/O assembly 8 (IB8), type:

CODE EXAMPLE 2-23 `flashupdate` Example Updating I/O Assembly 8

```
schostname:SC> flashupdate -f ftp://host/path ib8
```

Reboot the domain containing the I/O assembly that was updated in order to take advantage of the new features and bug fixes.

- To update the system controller real time operating system (`rtos`), perform this procedure from the platform console or watch the console output. Type:

CODE EXAMPLE 2-24 `flashupdate` Example Updating the System Controller Real Time Operating System

```
schostname:SC> flashupdate -f ftp://host/path rtos
```

This reboots the system controller.

help

Without arguments, lists currently available system controller commands. When an argument is supplied, displays command usage of what you specified and a short description.

Scope

platform shell, domain shell

Syntax

`help` [*command_name*] | [*partial_command_name*]

`help -h`

Options / Parameters

`-h` displays help for this command.

command_name is the name of the system controller command.

partial_command_name can be one letter of the command or a portion of the command name, such as `show`.

Description

The `help` command, without arguments, lists currently available commands in the system controller. When an argument is supplied, the `help` command displays a list of commands that begin with the specified argument. If only one command if found, full help is displayed. Otherwise, a short description is displayed for each command, beginning with the specified argument.

The `help` command is shell sensitive. It displays help information for the current shell only.

See Also

Examples

To display help information on the addboard command::

CODE EXAMPLE 2-25 help Command Example Displaying Information on the addboard Command

```
schostname:SC> help addboard

addboard -- assign a board to a domain

Usage: addboard -d <domain> <board> ...
       addboard -h

    -d -- the domain to assign the board to
    -h -- display this help message
```

To display all commands in the platform shell beginning with show:

CODE EXAMPLE 2-26 help Command Example Displaying Commands Beginning With the Word show

```
schostname:SC> help show

showboards       -- show board information
showcomponent    -- show state of a component
showdate         -- show the current date and time for the domain
showenvironment  -- show environment sensors
showkeyswitch    -- show the keyswitch position
showlogs         -- show the logs
```

To display all commands in the platform shell beginning with the letter a:

CODE EXAMPLE 2-27 help Command Example Displaying Commands Beginning With the Letter A

```
schostname:SC> help a

addboard -- assign a board to a domain

Usage: addboard -d <domain> <board> ...
       addboard -h

    -d -- the domain to assign the board to
    -h -- display this help message
```

history

Shows the command history with date and time stamps.

Scope

platform shell, domain shell

Syntax

```
history [-h]
```

Options / Parameters

`-h` displays help for this command.

Description

Shows the command history with date and time stamps for when the commands were executed. This command is shown for your shell and displays the last twenty commands only.

See Also

Example

CODE EXAMPLE 2-28 `history` Command

```
schostname:SC> history
1 : showboards         May 07 10:39:14
2 : showdate           May 07 10:39:30
3 : history            May 07 10:39:47
```

password

Sets the password for the platform or the domain.

Scope

platform shell, domain shell

Syntax—Platform Shell

`password [-d` *domainID*`]`

`password -h`

Syntax—Domain Shell

`password`

`password -h`

Options / Parameters

`-h` displays help for this command.

`-d` *-domainID* is domain a, b, c, or d.

Description

Sets the password for the platform or the domain. There are separate passwords for each domain and for the platform. If you set a password, entering the password is required for access.

It is very important to set the password for the platform and each domain even if a domain is not being used. This prevents people from creating and activating unauthorized domains.

Prior to allowing the password to be changed, the current password will be authenticated. Changed passwords take effect immediately. The old password will no longer be accepted.

You can remove the password by pressing Return at the `Enter new password` and `Enter new password again` prompts.

OpenBoot PROM passwords are different from the platform and domain passwords. The OpenBoot PROM in each domain supports the OpenBoot PROM security mode, which is a standard feature of the OpenBoot PROM software. For more information on the OpenBoot PROM security mode password, see your OpenBoot PROM documentation.

See Also

Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual (for a discussion of security).

Examples

You will see the following prompt (CODE EXAMPLE 2-29), when a password is already set and you type the `password` command at either the platform shell or the domain shell.

CODE EXAMPLE 2-29 `password` Command Example With No Password Set

```
schostname:SC> password
Enter new password:
Enter new password again:
schostname:SC>
```

If currently there is not a password assigned to the shell you are entering, you will not be prompted for the current password (CODE EXAMPLE 2-30).

CODE EXAMPLE 2-30 `password` Command Example With a Password Already Set

```
schostname:SC> password
Enter current password:
Enter new password:
Enter new password again:
schostname:SC>
```

poweroff

Powers off a component (power grid, power supply, board, fan tray), or a list of components.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
poweroff [-y|-n] all|grid#|device_name [device_names . . . ]
poweroff -h
```

Syntax for the Domain Shell

```
poweroff [-y|-n] all|device_name [device_names . . . ]
poweroff -h
```

Options / Parameters

`-h` displays help for this command.

`-y` will answer yes to any questions. This option is potentially hazardous. You can forcefully power off a component with the `-y` option.

`-n` answers no to any questions. You cannot forcefully power off a component with the `-n` option.

For a list of devices you power off, see the “Description” section.

Description

Powers off a component or a list of components. You *must* specify *device_name*(s).

The components that can be powered off in the platform shell are:

- `all` turns off all currently controllable boards.
- Power grid (grid0, grid1). The Sun Fire 6800 system has two power grids: `grid0` and `grid1`. Grid 1 controls power supplies ps3, ps4, and ps5. All other mid-range systems have one power grid, grid0. This grid controls power supplies ps0, ps1, and ps2.
- Power supply (ps0 - ps5)
- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)
- Repeater board (rp0 - rp3)
- Fan tray (ft0 - ft3)

The components that can be powered off in a domain shell are:

- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)

For the domain shell, the specified board must be in the current domain.

All slots listed by `showboards` can be powered off except for system controllers (SSC0 and SSC1) and the ID board. A powered-off component will not be monitored. Use the `showboards` command to display the power status of each board.

You can power off any component except when the board is in the Active state (see the `showboards` command to display the board state). When a component is in the *Active* state, you are asked to confirm with a warning of the consequences. If a board is active, a warning is displayed on the console advising you that the entire domain will go down.

If a board is active in a domain and you forcefully power off a device (board), the keyswitch for the domain will be set to standby. To change the keyswitch setting, use the `setkeyswitch` command.

Repeater boards do not have an Active state. You cannot power off a Repeater board because it is being used by the domain.

If the power supplies are needed to supply power to the boards and fans, you cannot power off the power supplies. If you power off a power supply, there might be insufficient power to support the current load.

If the fan trays are needed to cool the system, you cannot power them off because powering off fan trays result in insufficient cooling of the system.

See Also

`poweron, setkeyswitch, showboards`

Examples

To power off CPU/Memory board, `sb2` from the platform shell, type:

CODE EXAMPLE 2-31 `poweroff` Command Example Showing Powering Off CPU/Memory Board 2

```
schostname:SC> poweroff sb2
```

To power off all power supplies and currently controllable boards, type:

CODE EXAMPLE 2-32 `poweroff` Command Example Showing Powering Off All Power Supplies and Currently Controllable Boards in the Platform Shell

```
schostname:SC> poweroff all
```

To power off all CPU/Memory boards and I/O assemblies in domain A, type:

CODE EXAMPLE 2-33 `poweroff` Command Example Showing Powering Off All CPU/Memory Boards and I/O Assemblies in Domain A

```
schostname:A> poweroff all
```

To power off I/O assembly, `ib7` from domain A, type:

CODE EXAMPLE 2-34 `poweroff` Command Example Showing Powering Off I/O Assembly 7

```
schostname:A> poweroff ib7
```

Note – From the domain shell, you cannot power off power supplies, fan trays, Repeater boards, or power grids. To power off these components, connect to the platform shell.

---

poweron

Powers on a component (power grid, power supply, board, fan tray, or a list of components).

Scope

platform shell, domain shell

Syntax for the Platform Shell

`poweron all|`*grid#*`|`*device_name* `[`*device_names* `. . . ]`

`poweron -h`

Syntax for the Domain Shell

`poweron all|`*device_name* `[`*device_names* `. . . ]`

`poweron -h`

Options / Parameters

`-h` displays help for this command.

For the grids and devices you can power on, see the “Description” section.

Description

Powers on a device or a list of devices You *must* specify *device_name*(s).

The components that can be powered on in the platform shell are:

- `all` is all power supplies and boards.
- Power grid (grid0, grid1). The Sun Fire 6800 system has two power grids: grid0 and grid1. Grid 1 controls power supplies ps3, ps4, and ps5. All of the other mid-range systems have one power grid, `grid0`. All other mid-range systems have one power grid, grid0. This grid controls power supplies ps0, ps1, and ps2.
- Power supply (ps0 - ps5)
- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)
- Repeater board (rp0 - rp3)
- Fan tray (ft0 - ft3)

The components that can be powered on in a domain shell are:

- `all` powers on all CPU/Memory boards and I/O assemblies in the domain
- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)

For the domain shell, the board status must be assigned to the current domain. You can power on any component except when the board is in the Active state (see the `showboards` command to display the board state). When a component is in the *Active* state, you are asked to confirm with a warning of the consequences. Repeater boards do not have an Active state.

The fan trays power on automatically if any power supply is on. With the Sun Fire 6800/4810/4800 systems the power supplies power on automatically if a power supply in the same power grid is on. With the Sun Fire 3800 system, if grid power is on and the power supply switch is turned to the on position and the power cord is connected, the power supply powers on automatically.

See Also

`poweroff, showboards`

Examples

To power on CPU/Memory board, `sb2` from the platform shell, type:

TABLE 2-14 poweron Command Example Showing Powering On CPU/Memory Board 2

schostname:SC> **poweron sb2**

To power on I/O assembly, `ib7`, type:

TABLE 2-15 poweron Command Example Showing Powering On I/O Assembly 7

schostname:SC> **poweron ib7**

To power on CPU/Memory boards and I/O assemblies in the domain, type:

TABLE 2-16 poweron Command Example Showing Powering On All CPU/Memory Boards and I/O Assemblies in Domain A

schostname:A> **poweron all**

Note – From the domain shell, you cannot power on power supplies, Repeater boards, fan trays, or power grids. To power on these components, use the platform shell.

---

reboot

Reboots the system controller.

Scope

platform shell

Syntax

```
reboot [-y|-n]
```

```
reboot -h
```

Options / Parameters

`-y` does not prompt for confirmation.

`-n` does not execute this command, confirmation is requested.

`-h` displays help for this command.

Description

Reboots the system controller. The `reboot` command will not change any configuration settings. Active domains will continue to run.

If you reboot the system controller while the keyswitch operation is in progress, the keyswitch is set to `standby` when the system controller completes rebooting.

See Also

Example

To reboot the system controller, type:

```
schostname:SC> reboot
```

To reboot the system controller with the -n option, type:

```
schostname:SC> reboot -n
Are you sure you want to reboot the System Controller now? no (-n)
```

The `-n` option prompts you with the `Are you sure . . .` prompt to answer. Type `y` if you are sure you want to reboot the system controller. Type `n` if you do not want to reboot the system controller at this time.

reset

Resets the domain (in a domain shell) or the other system controller (in the platform shell).

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
reset [-y|-n] sscx
reset -h
```

Syntax for the Domain Shell

Note – In the domain shell, typing `reset`, without options, is the same as typing `reset -x`.

```
reset [-y|-n][-x|-a]
reset -h
```

Options / Parameters—Platform Shell

`-y` answers yes to the question asked (executes the command).

`-n` answers no to the question asked (does not execute the command).

`-h` displays help for this command.

`sscx` *(x* is 0 or 1) performs a hardware reset of the other system controller.

Caution – The second System Controller board is installed in the system for clock failover. Do not use the second System Controller board unless it is necessary to reset the primary System Controller board.

Options / Parameter—Domain Shell

`-h` displays help for this command.

`-x` resets via XIR (externally initiated reset). Uses XIR to try to obtain diagnostic data (default).

`-a` reset is equivalent to the OpenBoot PROM `reset-all` command.

`-y` resets even if the domain is active.

`-n` does not reset if the domain is active.

Description—Platform Shell

To perform software reset on the other System Controller board, use ssc1 if you are using ssc0. If you are using ssc1, use ssc0. Clock failover will be automatically enabled after the reset is complete. Messages will be displayed on the platform console.

Description—Domain Shell

Resetting the domain is not allowed if the keyswitch is in the secure, off, or standby positions.

By default, `reset` uses XIR (externally initiated reset) to reset the CPU processors in the domain. The externally initiated reset forces control of the domain into the OpenBoot PROM and begins the OpenBoot PROM error reset recovery actions. The error reset recovery actions preserve most domain states to allow collecting data needed for debugging the hardware and software, including a Solaris operating environment core file. The OpenBoot PROM error reset recovery actions are controlled by setting the OpenBoot PROM `error-reset-recovery` configuration variable. For the definition of this variable and the various settings, see TABLE 2-24 in Section , “`setupdomain`” on page 2-85.

You cannot reset a domain that has been paused. The domain is paused automatically when hardware detects an error. Messages on the domain console indicate that the domain is paused. To rectify this situation, turn the keyswitch off with `setkeyswitch off`. Then, turn the keyswitch on with `setkeyswitch on`.

See Also

`setkeyswitch`, `setupdomain`, `showdomain`, `showkeyswitch`, *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for step-by-step procedures on recovering from a hung domain or hard hung domain).

Examples

To reset the domain (from a domain shell), type::

CODE EXAMPLE 2-35 `reset` Command Example From Domain A

```
schostname:A> reset
```

This command uses XIR (externally initiated reset) to try to obtain diagnostic data.

To perform the equivalent operation as the OpenBoot PROM `reset-all` command, type:

CODE EXAMPLE 2-36 `reset` -a Command Example From Domain A

```
schostname:A> reset -a
```

To reset system controller `ssc0` from `ssc1`, type:

CODE EXAMPLE 2-37 `reset` Command Example Resetting `ssc0` From `ssc1` in the Platform Shell

```
schostname:SC> reset ssc0
```

restoreconfig

Restores the platform and domain configurations from a server.

Scope

platform shell

Syntax

`restoreconfig [-y|-n] -f` *url*

`restoreconfig -h`

Options / Parameters

`-h` displays help for this command.

`-y` does not prompt for confirmation.

`-n` does not execute this command is confirmation is required.

`-f` specifies a URL to restore from.

url is the directory containing the data files. The supported protocols are `ftp` and `http`. For example:

`ftp://[`*userid*`:`*password*`]@`*hostnam*`/`*path*

`ftp://`*hostname*`/`*path*

`http://[`*userid*`:`*password*`]@`*hostnam*`/`*path*

`http://`*hostname*`/`*path*

Note – The hostname you enter must be a complete host name (not a partial one).

Description

Restores the platform and domain configurations based on two data files: *schostanme*`.nvci` and *schostname*`.tod` at the specified URL. This process *requires* that the Solaris operating environment must be halted in each domain. Also, all domains must be powered off with the `setkeyswitch off` command.

Use this command when the System Controller board failed and after a working System Controller board is installed. You can also use this command when you want to restore the previous system controller configuration.

When the system controller configuration is restored, the system controller is rebooted. Set the date and time for the platform and for each domain using the `setdate` command.

The domains should be restored to their state when the configuration was stored. For example, if a domain was active, it should be activated immediately. After running the `restoreconfig` command, check the state of the domains.

See Also

`dumpconfig, setkeyswitch, setdate`

Example

CODE EXAMPLE 2-38 shows an example of the `restoreconfig` command with the required `-f` option

CODE EXAMPLE 2-38 `restoreconfig` Example

```
schostname:SC> restoreconfig -f ftp://host/path

The system controller will be rebooted when the restore is complete. The date
will need to be set in the platform and each domain.

Do you want to restore the system controller configuration now [no] yes

Retrieving: ftp://host/path/schostname.nvci
Retrieving: ftp://host/path/schostname.tod
Verifying data
NVCI has been restored
TOD has been restored

Restore complete. The system controller is being rebooted.
The date will need to be set in the platform and each domain.

Software Reset . . .
```

resume

Exits the domain shell and resumes access to the domain console.

Scope

domain shell

Syntax

```
resume
resume -h
```

Options / Parameters

`-h` displays help for this command.

Description

Exits the domain shell and resumes access to the domain console.

See Also

Example

CODE EXAMPLE 2-39 `resume` Command Example in Domain A

```
Domain A Shell

schostname:A> showboards

Slot   Pwr  Component Type State       Status       Domain
----   ---  -------------- ------      ------       ------
SB0    On   CPU Board      Active      Passed       A
IB6    On   PCI I/O Board  Active      Passed       A
schostname:A> resume
```

setdate

Sets the date, time, and time zone for the platform and for domains.

Scope

platform shell, domain shell

Syntax

`setdate [-v] [-t` *tz*`] [`*mmdd*`]`*HHMM*`[[`*cc*`]`*yy*`[.`*SS*`]`

`setdate [-v] [-r` *datehost*`]`

`setdate [-v] -t` *tz*

`setdate [-v] -t GMT`<+|->*offset from GMT (*TABLE 2-17*)*

`setdate -h`

Options / Parameters

`-t` *tz* sets the time zone using the time zone abbreviation (see TABLE 2-17). Only non-daylight savings time zones can be set. If you are in an area with daylight time or summer time, this is set automatically.

`-t GMT`<+|->*offset* from Greenwich Mean Time—*GMT* (see TABLE 2-17).

mm = month number, *dd* = day number in the month, *HH* = hour number (24-hour system), *MM* = minute number, *cc* = century minus 1, *yy* = last two digits of the year number, and *SS* = second number. For the century minus 1 value, use 21 (Twenty first century) minus 1, which yields 20.

`-r` *datehost* sets the current time using `rdate`. The host must be a valid system.

`-v` is verbose mode. Displays detailed information about the time zone that is set.

`-h` displays help for this command.

TABLE 2-17 Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time

Time Zone Abbreviation	Time Zone Name	Offset From Greenwich Mean Time (GMT)
ACT	Australian central time	GMT+9.5
AET	Australian eastern time	GMT+10
AGT	Argentina standard time	GMT-3
ART	Arabic (Egypt) standard time	GMT+2
AST	Alaska standard time	GMT-9
BET	Brazil eastern time	GMT-3
BST	Bangladesh standard time	GMT+6
CAT	Central African time	GMT+2
CNT	Canada Newfoundland time	GMT-3.5
CST	Central standard time	GMT-6
CTT	China Taiwan time	GMT+8
EAT	Eastern African time	GMT+3
ECT	European central time	GMT+1
EET	Eastern European time	GMT+2
EST	Eastern standard time	GMT-5
HST	Hawaii standard time	GMT-10
IET	Indiana eastern standard time	GMT-5
IST	India standard time	GMT+5.5
JST	Japan standard time	GMT+9
MET	Middle East time	GMT+3.5
MIT	Midway Islands time	GMT-11
MST	Mountain standard time	GMT-7
NET	Near East time	GMT+4
NST	New Zealand standard time	GMT+12
PLT	Pakistan Lahore time	GMT+5
PNT	Phoenix standard time	GMT-7
PRT	Puerto Rico and U S. Virgin Islands time	GMT-4

TABLE 2-17 Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time *(Continued)*

Time Zone Abbreviation	Time Zone Name	Offset From Greenwich Mean Time (GMT)
PST	Pacific standard time	GMT-8
SST	Solomon standard time	GMT+11
UTC	Universal Time Coordinated	GMT+0
VST	Vietnam standard time	GMT+7

Description

Sets the date and time for the platform and domains. This command, when invoked from the platform, will have no effect on the date and time in each domain and vice versa. You can set up to five different times and time zones; one time and time zone for the platform and different times and time zones for each of the four domains.

If your time zone area is using daylight or summer time, this is set automatically.

Note – You cannot set the date from the system controller in a domain while the Solaris operating environment is running. To set the date while the Solaris operating environment is running, use the Solaris operating environment `date` command.

See Also

`showdate`

Examples

When you type the `setdate` command from the platform, this sets the date and time for the platform. When you type the `setdate` command from the domain, this sets the date and time for the domain.

To set the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds, type:

TABLE 2-18 `setdate` Command Example in the Platform Shell

```
schostname:SC> setdate 042018152001.10
Thu Apr 20 18:15:10 PST 2001
```

To set the date from a date host:

CODE EXAMPLE 2-40 `setdate -r` Command Example Setting the Date From a Date Host

```
schostname:SC> setdate -r datehost
```

To set the time zone to Pacific Standard Time (PST), using the offset from Greenwich mean time—GMT, and the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds (TABLE 2-17), type:

CODE EXAMPLE 2-41 `setdate -t` Command Example Setting the Time Zone to Pacific Standard Time Using the Offset From Greenwich Mean Time

```
schostname:SC> setdate -t GMT-8 042018152001.10
Thu Apr 20 18:15:10 PST 2001
```

To set the time zone to Eastern Standard Time (EST), using the time zone abbreviations, and the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds (TABLE 2-17), type:

CODE EXAMPLE 2-42 `setdate -t` Command Example Setting the Time Zone to Eastern Standard Time Using Time Zone Abbreviations and Also Setting the Date

```
schostname:SC> setdate -t EST 042018152001.10
Thu Apr 20 18:15:10 EST 2001
```

To set just the time zone for Japan Standard Time using offsets from Greenwich Mean Time—GMT (TABLE 2-17) and *not* the date and time, type:

CODE EXAMPLE 2-43 `setdate -t` Command Example Setting the Time Zone to Japan Standard Time Using the Offset From Greenwich Mean Time

```
schostname:SC> setdate -t GMT+9
Thu Apr 20 18:15:10 GMP+9 2001
```

To set just the time zone for European Central Time using the time zone abbreviations (TABLE 2-17) and *not* the date and time, type:

CODE EXAMPLE 2-44 `setdate -t` Command Example Setting the Time Zone to Eastern Central Time Using Time Zone Abbreviations

```
schostname:SC> setdate -t ECT
Thu Apr 20 18:15:10 ECT 2001
```

setdefaults

Sets the default configuration values.

Caution – This is a destructive command. Use with caution and care.

Scope

platform shell, domain shell

Syntax

```
setdefaults [-y|-n] [-a]
setdefaults -h
```

Options / Parameters

`-y` does not prompt for confirmation. The option reboots the system and sets the defaults.

`-n` does not execute the command if confirmation is requested.

`-a` adds missing parameters.

`-h` displays help for this command.

Description

This command does the following:

- Sets the system controller to the default values. When invoked for a domain, this command sets the defaults for that domain only. When invoked from the platform, this command sets the defaults for all domains and the platform.
- Erases all platform and domain configurations (platform).
- Removes passwords (domain and platform).

Caution – This command requires that all domains are inactive (not running the OpenBoot PROM, POST, or the Solaris operating environment) and the virtual keyswitch be set to off.

See Also

Example

CODE EXAMPLE 2-45 `setdefaults` Example

```
schostname:SC> setdefaults

WARNING!

You are about to reset all configuration data to default values.
All domain configurations, ACLs, users and data buffers will be
lost.

The System Controller will be REBOOTED after the defaults are set.

Do you want to restore the default values and reboot now? [no]
```

If you answer yes, the system controller is rebooted after the defaults are set.

If you answer no, the default values are not restored.

If a password is set for the platform shell, the password will be required.

setkeyswitch

Changes the position of the virtual keyswitch to the specified value.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
setkeyswitch [-y|-n] -d domainID off
setkeyswitch -h
```

Syntax for the Domain Shell

```
setkeyswitch [-y|-n] [off|standby|on|diag|secure]
setkeyswitch -h
```

Options / Parameters

`-h` displays help for this command.

`-y` does not prompt for confirmation.

`-n` does not execute this command is confirmation is requested.

`-d` *domainID* (platform shell only) is the domain ID of the domain you want power off (`a`, `b`, `c`, or `d`).

For `setkeyswitch` parameters and descriptions, see TABLE 2-19.

TABLE 2-19 `setkeyswitch` Values and Description

Value	Description
`off`	Changes the board state of all the boards belonging to a domain to the Assigned board status and places the boards in low-power mode, which allows you to remove the boards from the system. The domain will not be initialized at system power-on.
`on`	Powers on and initializes the domain. The system controller brings the domain into OpenBoot PROM through POST. If the OpenBoot PROM has `auto-boot?` set to `true`, then the Solaris operating environment boots automatically.
`standby`	Changes the board state of all the boards belonging to a domain to the Assigned board state and turns on the boards. The domain will not be initialized at system power on, but the boards that comprise the domain will be powered on.
`diag`	Similar to the `setkeyswitch on` command except the POST verbose mode is set to on and the POST diagnostic level is set to maximum.
`secure`	Similar to the `setkeyswitch on` command except that the `break` command and the `reset` commands are ignored. CPU/Memory board and I/O assembly flash PROM updates are not allowed.

TABLE 2-20 shows the results when you change a keyswitch setting.

TABLE 2-20 Results of Changing the Keyswitch From the Current Setting to a New Setting

Current Keyswitch	New Keyswitch Setting	Result
off	off	Keyswitch position not changed.
off	standby	Changes the board state of all the boards belonging to a domain to Assigned and powers on all of the boards. The domain will not be initialized at system power on.
off	on	Powers on and initializes the domain. The domain will go through POST and the OpenBoot PROM. The Solaris operating environment boots automatically unless you have set the OpenBoot PROM `auto-boot?` set to false.
off	diag	Similar to on, except POST verbose and diagnostic level is to set to maximum. The domain will boot from the diag-device.
off	secure	Similar to on, except the reset and break commands are ignored by the domain shell.
on	secure	`break` and `reset` are ignored.

TABLE 2-20 Results of Changing the Keyswitch From the Current Setting to a New Setting *(Continued)*

Current Keyswitch	New Keyswitch Setting	Result
secure	on	`break` and `reset` are not ignored.
standby	on	Similar to setting the virtual keyswitch from off to on except that you do not have to wait for the system controller to turn on the boards and prepare them.
diag	on	Has no effect on a running domain. This operation only changes the keyswitch position and does not perform any other function since the domain is already running.
standby secure on diag	off	All boards will be powered off (set to low -power mode and all monitoring of the boards will stop.

Description

Each domain has a virtual keyswitch with five positions: off, standby, on, diag, and secure. The `setkeyswitch` command changes the position of the virtual keyswitch to the specified value. The virtual keyswitch replaces the need for a physical keyswitch for each domain.

This command is also available in the platform shell. This command is meant to be used in the platform shell in case of problems with a domain. From the platform shell, you can only turn the virtual keyswitch off. This action shuts down the domain. Turn the keyswitch off from the domain, if possible.

If the domain is running the Solaris operating environment, the `setkeyswitch` command parameters `off` and `standby` require confirmation.

If the domain is already powered on, the `setkeyswitch on`, `setkeyswitch diag`, and `setkeyswitch secure` commands will change only the position of the virtual keyswitch. If components making up a domain such as a I/O assembly or CPU/Memory board need to be powered on, the system attempts to power on these boards or devices also.

If a failure occurred during keyswitch transition, it will be reset to `off` or to `standby` after recovering from the failure.

The system controller maintains the position of each keyswitch even during power failures or physical power cycles.

See Also

`showkeyswitch`

Examples

To turn the virtual keyswitch on and power on the domain, type:

TABLE 2-21 `setkeyswitch on` Example

schostname:A> **setkeyswitch on**

To shut down the domain by turning the virtual keyswitch off, type:

TABLE 2-22 `setkeyswitch off` Example

schostname:A> **setkeyswitch off**

To shut down domain B from the platform shell type:

TABLE 2-23 `setkeyswitch off` Example Shutting Down Domain B From the Platform Shell

schostname:SC> **setkeyswitch -d b off**

setupdomain

Sets up domain specific parameter values and configures the domain.

Scope

domain shell

Syntax

`setupdomain [-p` *part*`]...`

`setupdomain -h`

Options / Parameters

`-h` displays help for this command.

`-p` *part* configures the specified information and has the following options:

- `bootparams` configures the boot parameter information.
- `loghost` configures the log hosts— *loghost*`[:`*facility*`]`
- `snmp` configures SNMP information.

TABLE 2-24 describes each parameter value.

Description

Sets up domain specific values. Configures the domain. You can configure each domain differently and independently.

See Also

`setdate, showdate, showdomain`

Example

CODE EXAMPLE 2-46 Variables for the `setupdomain` Command

schostname:A> **`setupdomain`**

```
Domain Boot Parameters
----------------
diag-level [off]:
verbosity-level [min]:
error-level [max]:
interleave-scope [within-board]:
interleave-mode [optimal]:
reboot-on-error [false]:
OBP.use-nvramrc? [<OBP default>]:
OBP.auto-boot? [<OBP default>]:
OBP.error-reset-recovery [<OBP default>]:

Log hosts
---------
```

`Loghost [ ]:` *The name or the IP address of the loghost for this domain. The format is loghost*`:[`*facility*`].` *Note that facility is defined by the Solaris operating environment* `systlogd` *in* `/etc/syslog.conf.`

```
SNMP
-----
```

`Domain Description[]:` *A brief description for this domain (its function).*
`Domain Contact[]:` *The name of the primary domain administrator.*
`Trap Hosts [( )]:` *The IP address of the loghost for this domain*
`Public Community String [ ]:` *The authentication string for SNMP readers. .*
`Private Community String [ ]:` *Type the authentication string for readers and writers.*

If you enter an invalid value, the system controller displays “invalid entry” and prompts you to enter a proper value (TABLE 2-24). If a value is not specified, it retains its current value. When an invalid entry is entered, it also lists the valid responses. Typing a dash (-), clears the entry.

Current values are displayed in [ ]. TABLE 2-24 lists the `setupdomain` parameter values.

TABLE 2-24 Parameter Values for the `setupdomain` Command

Parameter	Value	Description
diag-level	`default` (default value)	All system board components are tested with all tests and test patterns, except for memory and Ecache modules. For memory and Ecache modules, all locations are tested with multiple patterns. More extensive, time-consuming algorithms are not run at this level.
	`init`	Only system board initialization code is run. No testing is done. This is a very fast pass through POST.
	`quick`	All system board components are tested using few tests with few test patterns.
	`mem1`	Runs all tests at the default level plus more exhaustive DRAM and SRAM test algorithms.
	`mem2`	This is the same as `mem1` with the addition of a DRAM test that does explicit compare operations of the DRAM data.
verbosity-level	`off`	No status messages are displayed.
	`min` (default value)	Test names status messages, and error messages are displayed.
	`max`	Subtest trace messages are displayed.
error- level	`off`	No error messages are displayed.
	`min`	The failing test name is displayed.
	`max` (default value)	All relevant error status is displayed.
interleave-scope	`within-board` (default value)	The memory banks on a system board will be interleaved with each other.
	`across-boards`	The memory will be interleaved on all memory banks across all of the boards in the domain.
interleave-mode	`optimal` (default value)	The memory is mixed-size interleaving in order to gain optimal performance.
	`fixed`	The memory is fixed-size interleaving.
	`off`	There is no memory interleaving.
reboot-on-error	`false` (default value)	The domain will be paused when there is an error.
	`true`	The domain will be rebooted.

TABLE 2-24 Parameter Values for the `setupdomain` Command *(Continued)*

Parameter	Value	Description
OBP.use-nvramrc?		This parameter is the same as the OpenBoot PROM `nvramrc?` parameter. This parameter uses aliases that are stored in `nvramrc`.
	`true`	The OpenBoot PROM executes the script stored in `nvramrc` if this parameter is set to true.
	`false` (default value)	The OpenBoot PROM does not evaluate the script stored in `nvramrc` if this parameter is set to false.
OBP.auto-boot?		Controls booting of the Solaris operating environment.
	`true` (default value)	If this value is true, the domain boots automatically after POST has run.
	`false`	If this parameter value is set to false, you will obtain the OpenBoot PROM `ok` prompt after POST runs, from which you must type a `boot` command to boot the Solaris operating environment.
OBP.error-reset-recovery		Controls the behavior of the domain after an externally initiated reset (XIR) as well as a red mode trap.
	`sync` (default value)	The OpenBoot PROM invokes `sync`. A core file is generated. If the invocation returns, the OpenBoot PROM performs a reboot.
	`none`	The OpenBoot PROM prints a message describing the reset trap that triggered the error reset and passes control to the OpenBoot PROM `ok` prompt. The message describing the reset trap type is platform specific.
	`boot`	The OpenBoot PROM firmware reboots the domain. A core file is not generated. Rebooting a domain occurs using the OpenBoot PROM settings for `diag-device` or `boot-device`, depending on the value of the OpenBoot PROM configuration variable `diag-switch?` If `diag-switch?` is set to true, the device names in `diag-device` will be the default for boot. If `diag-switch?` is set to false, the device names in `boot-device` will be the default for boot.
Log Host		The name or the IP address of the loghost for this domain (SNMP). The format is *loghost*:[*facility*]. Note that facility is defined by the Solaris operating environment `systlogd` in `/etc/syslog.conf`.

TABLE 2-24 Parameter Values for the `setupdomain` Command *(Continued)*

Parameter	Value	Description
Domain Description		Enter a brief description for this domain. For example, its function.
		Enter the name of the primary domain administrator.
Trap Hosts		Enter the name or the IP address of the trap host for this domain.
Public Community String		Authentication string for SNMP readers. The default value is A-public, B-public, and so on. **Note**—Make sure that the value in this field is the same as the Sun Management Center software for the Sun Fire 6800/4810/4800/3800 systems value for public community string.
Private Community String		Authentication string for readers and writers. The default value is A-private, B-private, and so on. **Note**—Make sure that the value in this field is the same as the Sun Management Center software for the Sun Fire 6800/4810/4800/3800 systems value for private community string.

setupplatform

Configures the platform specific variables.

Scope

platform shell

Syntax

`setupplatform [-p` *part*`]. . .`

`setupplatform -h`

Options / Parameters

`-h` displays help for this command.

`-p` *part* . . . configures the specified part and has the following options:

- `acls` is ACLs (Access Control List)
  - `-` clears the boards from the ACL
  - `+` adds all boards to the ACL
  - `-a` *board* . . . adds a board
  - `-d` *board* . . . deletes a board
- *loghost* is the IP address or hostname of the syslog loghost in the format *loghost*`[:`*facility*`]` (The hostname you enter must be a complete host name (not a partial one).
- `network` is the network settings
- `partition` sets the partition mode to single (one partition) or dual (two partitions)
- `sc` is the system controller POST diag level
- `snmp` is SNMP configuration (The hostname you enter must be a complete host name (not a partial one).

Description

Sets up platform specific values. For the parameters you can configure, see TABLE 2-25.

TABLE 2-25 `setupplatform` Parameter Values

Parameter	Description
`Is the System Controller on a Network?`	If the answer is no, leave all network parameters as unconfigured
`Use DHCP or static network settings?`	• `DHCP` means the system controller network configuration is retrieved from a DHCP server. • `Static` means the network settings will be manually configured. If you select the static setting, you will be asked for the following network parameters.
`Hostname` (Network parameter)	The human readable network identity for this system controller.
`IP address` (Network parameter)	The network identity used by computers.
`Netmask` (Network parameter)	For this value, specify how much of the address should be reserved for subdividing networks into subnetworks. Must be specified in dot-notation address. For example `255.255.255.0`.
`Gateway` (Network parameter)	IP address of the gateway.
`DNS Domain` (Network parameter)	Domain name. For example, *xxx*`.`*xxx*`.com`.
`Primary DNS Server` (Network parameter)	IP address of your primary DNS server.
`Secondary DNS Server` (Network parameter)	IP address of your secondary DNS server.
`Loghost`	*IP address* or *hostname* of the system that will accept syslog messages. The format is *loghost*[:*facility*]. For example, *IP address*:local0. Errors, warnings, and some command will be logged to this system.
`SNMP Agent?`	The SNMP agent enables or disables SNMP.
`Platform Description`	Default value is the platform model name.
`Platform Contact`	The name of the person who is responsible for this system. This name will be used by SNMP. Note that if SNMP is not going to be enabled, this field is informational only.
`Platform Location`	Geographic location of the system. This information will be used by SNMP. Note that if SNMP is not going to be enabled, this field is informational only.
`Trap Hosts`	IP address or name of the trap host (SNMP).

TABLE 2-25 `setupplatform` Parameter Values *(Continued)*

Parameter	Description
`Public Community String`	Authentication string for SNMP readers. The default is `P-public`.
`Private Community String`	Authentication string for readers and writers. The default is `P-private`.
`ACL for Domain A`	Default value is to have all slots assigned to all domains. • Prefixing a slot name with `-d` deletes the slot from ACLs of this domain. • Typing a prefix of `-a` (add) adds a slot. • Typing a dash (-) clears the entire list. • Typing a + (plus sign) adds all. • Pressing the Return key does not change the value displayed inside [ ].
`ACL for Domain B`	See the description for ACL for Domain A.
`ACL for Domain C`	See the description for ACL for Domain A.
`ACL for Domain D`	See the description for ACL for Domain A.
`SC POST diag level`	The system controller boot parameters (diag level) can be set to: `off`, `min`, or `max`.
`off`	Only system board initialization code is run. No testing is done. POST is not run.
`min`	All system board components are tested using few locations with few test patterns.
`max`	All system board components are tested with all tests and test patterns.
`Configure platform for single or dual partition mode`	The two options are single or dual. The default is single. For a description of single and dual partition mode, refer to the "Overview" chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual.*

See Also

`setdate`, `setupdomain`, `showdate`, `showdomain`, `showplatform`, and the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for a step-by-step procedure on using `setupplatform`).

Example

CODE EXAMPLE 2-47 Output From `setupplatform` Command

```
schostname:SC> setupplatform

Network Configuration
---------------------
Is the System Controller on a network? [yes]
Network settings? [static]:
Hostname []:
IP Address []:
Netmask []:
Gateway []:
DNS Domain []:
Primary DNS Server []:
Secondary DNS Server []:

Loghost [ ]:

SNMP
----
Platform Description [Sun Fire 6800]: System type
Platform Contact [ ]: Platform administrator's name
Platform Location [ ]: Geographic system location
Enable SNMP Agent? [yes]:
Trap Hosts []: SNMP trap host IP address or name
Public Community String [P-public]:Authentication string for SNMP readers
Private Community String [P-private]: Authentication string for readers and
writers

ACLs
----
ACL for domain A [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain B [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain C [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain D [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:

SC POST
-------
SC POST diag level [min]:

Partition Mode
--------------
Configure platform for single or dual partition mode? [single]:
```

showboards

Displays the assignment information and status for all of components in the system.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`showboards [-ev] [-y|-n] [-d` *domainID*`] [-p` *part*`]`

`showboards -h`

Syntax for the Domain Shell

`showboards [-ev] [-y|-n] -d` *domainID*`] [-p` *part*`] . . .`

`showboards -h`

Options / Parameters

`-e` includes empty slots in the output.

`-v` shows boards including shared components and empty slots for the platform shell only. Verbose mode for parts.

`-d` *domainID* (platform shell only) is a, b, c, or d. Displays information for boards assigned to that domain.

`-p` *part* shows only a specific part and can be:

- `cpu` shows you CPU information.
- `proms` shows you the firmware revision of the PROMs.
- `memory` shows memory information.

`-y` answers yes to all questions.

`-n` answers no to all questions.

`-h` displays help for this command.

Description

Displays the assignment information and status for all of the components in the system. For example, CPU/Memory boards, I/O assemblies, fan trays, and so on. For the domain shell, the display shows components assigned to the domain and can also include boards that are included in the ACL, but not assigned to any other domain.

See Also

`addboard, deleteboard, testboard`

Examples—Platform Shell

CODE EXAMPLE 2-48 `showboards` Command for the Platform Shell

```
schostname:SC> showboards

Slot      Pwr    Component Type   State        Status       Domain
----      ---    --------------   -----        ------       ------
/N0/SB0   On     CPU Board        Active       Passed       A
/N0/SB2   On     CPU Board        Active       Passed       C
/N0/SB4   On     CPU Board        Active       Passed       D
/N0/IB6   On     PCI I/O Board    Active       Passed       A
/N0/IB7   On     PCI I/O Board    Active       Passed       C
/N0/IB8   On     PCI I/O Board    Active       Passed       D
```

The `showboards` command with the `-v` option displays the grid number.

CODE EXAMPLE 2-49 `showboards -v` Command for the Platform Shell

```
schostname:SC> showboards -v

Slot      Grd    Pwr    Component Type          State       Status        Domain
----      ---    ---    --------------          -----       ------        ------
SSC0      -      On     System Controller       -           Passed        -
SSC1      -      On     System Controller       -           -             -
ID0       -      On     Sun Fire 6800           -           -             -
                        Centerplane
PS0       0      On     A152 Power Supply       -           OK            -
PS1       0      On     A152 Power Supply       -           OK            -
PS2       0      -      Empty Slot              -           -             -
```

CODE EXAMPLE 2-49 `showboards -v` Command for the Platform Shell *(Continued)*

```
PS3       1      On     A152 Power Supply       -           OK           -
PS4       1      On     A152 Power Supply       -           OK           -
PS5       1      On     A152 Power Supply       -           OK           -
FT0       0,1    On     Fan Tray                Low Speed   OK           -
FT1       0,1    On     Fan Tray                Low Speed   OK           -
FT2       0,1    On     Fan Tray                Low Speed   OK           -
FT3       0,1    On     Fan Tray                Low Speed   OK           -
RP0       0      On     Repeater Board          -           OK           -
RP1       0      On     Repeater Board          -           OK           -
RP2       1      Off    Repeater Board          -           OK           -
RP3       1      Off    Repeater Board          -           OK           -
/N0/SB0   0      Off    CPU Board               Assigned    Not tested   A
/N0/SB1   1      On     CPU Board               Active      Passed       A
/N0/SB2   0      On     CPU Board               Assigned    Degraded     A
/N0/SB3   1      On     CPU Board               Assigned    Degraded     A
/N0/SB4   0      Off    CPU Board               Assigned    Not tested   A
/N0/SB5   1      On     CPU Board               Active      Passed       A
/N0/IB6   0      On     PCI I/O Board           Active      Passed       A
/N0/IB7   1      Off    PCI I/O Board           Assigned    Not tested   B
/N0/IB8   0      Off    PCI I/O Board           Assigned    Not tested   C
/N0/IB9   1      Off    PCI I/O Board           Assigned    Not tested   D
```

TABLE 2-26 Output Header Definitions for the `showboards -v` Command

Header	Description
Slot	Slot designator. The N*x* in the slot descriptor is the node number.
Grd	The power grids (in the Sun Fire 6800 system) or the power grid (in the other mid-range systems the system gets power from.
Pwr	Indicates if the power status of the device is off or on.
Component type	Component description.
State	Describes board state. Possible values are: Active, Assigned, Available and – (dash) (TABLE 2-3). The - board state means that the board state does not apply to this slot.
Status	Contains the result of the `testboard` command or POST. Possible values are Passed, Failed, Not Tested, Degraded, Under Test, and – (dash) (TABLE 2-6).

TABLE 2-26 Output Header Definitions for the `showboards -v` Command *(Continued)*

Header	Description
Domain	Indicates which domain the board belongs to: Possible values are: • **A-D—**The Sun Fire 6800 systems can have up to four active domains (A – D). The other mid-range systems can have up to four domains, but only two domains can be active domains. Active domains must contain an I/O assembly and a CPU/Memory board. For more information on domains, refer to the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual.* • **Isolated**—The board does not belong to a domain. • **–** (dash) The board cannot be assigned to a specific domain. It is a shared resource.

Example—Domain Shell

CODE EXAMPLE 2-50 `showboards` Command for the Domain Shell

```
schostname:A> showboards
Showing boards for domain A
Slot     Pwr   Component Type        State       Status       Domain
----     ---   --------------        ------      ------       ------

/N0/SB0  On    CPU Board             Active      Passed       A
/N0/SB2  Off   CPU Board             Assigned    Not Tested   A
/N0/SB4  Off   CPU Board             Assigned    Not Tested   A
/N0/IB6  On    PCI I/O Board (F4800) Active      Passed       A
```

showcomponent

Shows the state of a component.

Scope

platform shell, domain shell

Syntax

`showcomponent` *board_name*

`showcomponent -h`

Options / Parameters

`-h` displays help for this command. Includes *component_name* syntax.

- *boardname* for CPU / Memory boards is SB0 - SB5
- *boardname* for I / O assemblies is IB6 - IB9

Description

Shows the state of a component and details about the components on a board.

See Also

`enablecomponent, disablecomponent`, *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for a step-by-step procedure on displaying a component).

Examples

CODE EXAMPLE 2-51 shows sample output for the `showcomponent sb4` command. The abbreviations for the `Component` field are:

- N*x* Node name
- SB*x* CPU/Memory board, where x is 0 – 5
- P*x* Port, where x is 0 – 3
- B*x* Physical memory bank, where x is 0 – 1
- L*x* Logical memory bank, where x is 0 – 3

CODE EXAMPLE 2-51 `showcomponent sb4` Sample Output

```
schostname:SC> showcomponent sb4
Component          Status    Pending  POST       Description
---------          ------    -------  ----       -----------
/NO/SB4/P0         enabled   -        pass       UltraSPARCIII,750Mhz, 8M Cache
/NO/SB4/P1         enabled   -        pass       UltraSPARCIII,750Mhz, 8M Cache
/NO/SB4/P2         enabled   -        pass       UltraSPARCIII,750Mhz, 8M Cache
/NO/SB4/P3         enabled   -        pass       UltraSPARCIII,750Mhz, 8M Cache
/NO/SB4/P0/B0/L0   enabled   -        pass       256M DRAM
/NO/SB4/P0/B0/L2   enabled   -        pass       256M DRAM
/NO/SB4/P0/B1/L1   enabled   -        pass       256M DRAM
/NO/SB4/P0/B1/L3   enabled   -        pass       256M DRAM
/NO/SB4/P1/B0/L0   enabled   -        pass       256M DRAM
/NO/SB4/P1/B0/L2   enabled   -        pass       256M DRAM
/NO/SB4/P1/B1/L1   enabled   -        pass       256M DRAM
/NO/SB4/P1/B1/L3   enabled   -        pass       256M DRAM
/NO/SB4/P2/B0/L0   enabled   -        pass       256M DRAM
/NO/SB4/P2/B0/L2   enabled   -        pass       256M DRAM
/NO/SB4/P2/B1/L1   enabled   -        pass       256M DRAM
/NO/SB4/P2/B1/L3   enabled   -        pass       256M DRAM
/NO/SB4/P3/B0/L0   enabled   -        pass       256M DRAM
/NO/SB4/P3/B0/L2   enabled   -        pass       256M DRAM
/NO/SB4/P3/B1/L1   enabled   -        pass       256M DRAM
/NO/SB4/P3/B1/L3   enabled   -        pass       256M DRAM
```

where:

`Component` is the name of the board and its components.

`Status` is the status of the board or component.

`Pending`

`POST` is the status of POST (passed or failed).

`Description` is a description of the board or component.

CODE EXAMPLE 2-52 shows sample output for the `showcomponent ib6` command. The abbreviations for the Component field are:

- `N`*x* Node name
- `IB`*x* I/O assembly, where *x* is 6 – 9
- `P`*x* Port, where *x* is 0 – 1
- `B`*x* Bus, where *x* is 0 – 1
- `C`*x* Card in the I/O assembly, where *x* is 0 – 7

CODE EXAMPLE 2-52 `showcomponent ib6` Sample Output

```
schostname:SC> showcomponent ib6
Component          Status    Pending   POST      Description
---------          -------   -------   ----      -----------
/N0/IB6/P0         enabled   -         pass      IO controller 0
/N0/IB6/P1         enabled   -         pass      IO controller 1
/N0/IB6/P0/B1      enabled   -         untest    66/33MHz PCI Bus
/N0/IB6/P0/B0      enabled   -         untest    33MHz PCI Bus
/N0/IB6/P0/B1/C0   enabled   -         untest    33MHz 5V Short PCI card
/N0/IB6/P0/B1/C1   enabled   -         untest    33MHz 5V Short PCI card
/N0/IB6/P0/B1/C2   enabled   -         untest    33MHz 5V LOng/Short PCI card
/N0/IB6/P0/B1/C3   enabled   -         untest    33MHz 5V LOng/Short PCI card
```

showdate

Displays the current date and time for the platform or domain.

Scope

platform shell, domain shell

Syntax

```
showdate [-t] [-v]
showdate -h
```

Options / Parameters

`-h` displays help for this command.

`-t` lists available time zones.

`-v` is verbose mode.

Description

Displays the current date and time for the platform.

See Also

`setdate, showdate`

Example—Platform Shell

CODE EXAMPLE 2-53 `showdate` Command for the Platform Shell

```
schostname:SC> showdate
Mon Apr 03 12:31:40 EDT 2001
```

Example—Domain Shell

CODE EXAMPLE 2-54 `showdate` Command for the Domain Shell

```
schostname:A> showdate
Mon Apr 03 12:31:40 EDT 2001
```

showdomain

Displays the domain parameter settings, which is the status and non-configurable information for the current domain.

Scope

domain shell

Syntax

`showdomain [-v] [-p` *part*`]...`

`showdomain -h`

Options / Parameters

This command without any arguments displays all the parameter values configured by the `setupdomain` command.

`-v` is verbose mode.

`-p` *part* is a parameter for the information to display:

- `acls` shows the ACLs (Access Control List)
- `bootparams` shows boot parameter information
- `loghosts` shows the log hosts
- `mac` shows MAC addresses
- `snmp` shows SNMP configuration
- `status` shows domain status

`-h` displays help for this command.

Description

Displays the domain parameter settings, which is the status and non-configurable information for the current domain.

See Also

`setupdomain, showdate`

Example

To display parameters assigned by the `setupdomain` command, type `showdomain`.

CODE EXAMPLE 2-55 `showdomain` Command Example

```
schostname:A> showdomain

Domain  Solaris Nodename  Domain Status  Keyswitch
------  ----------------  -------------  ---------
A       -                 Powered Off    off

Domain Boot Parameters
----------------------
diag-level = off
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
OBP.use-nvramrc? = true
OBP.auto-boot? = false
OBP.error-reset-recovery = none

Loghost for Domain A:

SNMP Agent: enabled
Domain Description:
Domain Contact:
Trap Hosts:
Public Community String: P-public
Private Community String: P-private

ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

schostname:A>
```

To display the verbose output of parameters assigned by the `setupdomain` command, type `showdomain -v` (CODE EXAMPLE 2-56).

CODE EXAMPLE 2-56 showdomain -v Command Example

```
schostname:A> showdomain -v

Domain  Solaris Nodename  Domain Status  Keyswitch
------  ----------------  -------------  ---------
A       -                 Powered Off    off

Domain Boot Parameters
----------------------
diag-level = off
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
OBP.use-nvramrc? = true
OBP.auto-boot? = false
OBP.error-reset-recovery = none

             MAC Address      HostID
             -------------    ------
Domain A     xx:xx:xx:xx:xx:xx  xxxxxxxx

Loghosts
--------
Loghost for Domain A:

SNMP
----
SNMP Agent: enabled
Domain Description:
Domain Contact:
Trap Hosts:
Public Community String: P-public
Private Community String: P-private

SNMP packets received: x
SNMP packets sent: xx
SNMP traps sent: xxx

ACLs
----
ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
```

To display boot parameter information, type:

CODE EXAMPLE 2-57 `showdomain -p bootparams` Example Displaying Boot Parameter Information

```
schostname:A> showdomain -p bootparams

diag-level = off
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = false
OBP.use-nvramrc? = true
OBP.auto-boot? = false
OBP.error-reset-recovery = sync

schostname:A>
```

showenvironment

Displays the current environmental status, temperatures, currents, voltages, and fan status, for the system (platform shell) or the current domain (domain shell).

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
showenvironment [-ltvuw] [-d domainID] [-p part]
showenvironment [-ltvuw] device_name
showenvironment -h
```

Syntax for the Domain Shell

```
showenvironment [-ltvuw] [-p part]
showenvironment [-ltvuw] device_name
showenvironment -h
```

Options / Parameters

`-h` displays help for this command.

`-l` displays the limits that apply to each selected measurement. These values are the threshold for each measurement. Exceeding the threshold causes the status to display `Max` or `Min`.

`-d` *domainID* is a, b, c, or d (platform shell only). Displays information for boards assigned to that domain.

`-p` *part* shows a specific part (TABLE 2-27).

`-t` prints header titles.

`-u` updates data now. Polls all sensors for new values.

`-v` is verbose mode. This is the equivalent of the options `-l` and `-w`.

`-w` shows warning thresholds.

device_name is the name of the board. Displays information for this board only.

TABLE 2-27 `showenvironment -p` *part* Descriptions

Report	Description
currents	Currents (power supplies only)
faults	Displays only measurements that are above High or below Low
temps	Shows temperatures only.
voltage	Shows voltages only.
fans	Shows fan states.

Description

Displays the current environmental status, temperatures, currents, voltages, and fan status for the system (platform shell) or the current domain (domain shell). In order for this command to be recognized in a domain shell, the boards must be assigned to the domain and the boards must be powered on. If you are running the command from the domain shell only, system boards assigned to the domain are displayed.

This command also displays minimum and maximum allowable values for each component being monitored. Samples are updated periodically, but the operator can get new samples by providing the proper argument. If no arguments are supplied, all applicable environmental information will be displayed. TABLE 2-28 describes the `showenvironment` output headers.

TABLE 2-28 `showenvironment` Output Header Description

Header	Value	Description
Slot		Slot ID
Device		Device being monitored by the sensor
Sensor		Component that measures the environmental data of the device
Value		Value returned by the sensor (data was acquired seconds ago)
Units		Applicable unit for the sensor (for valid units, see the Value column)
	C	Celsius
	V	Volts

TABLE 2-28 `showenvironment` Output Header Description *(Continued)*

Header	Value	Description
	A	Amps
Age		Age of the reading being displayed (seconds)
Status		Values for Status. See the Value column
	max	Value exceeded Max threshold (fault condition)
	high	Value below High threshold (fault condition)
	low	Value exceeded Low threshold (warning condition)
	min	Value below Min threshold (warning condition)
	ok	Value is within range (within Low and High)
LoWarn		Displayed only with the `-v` and `-w` options. Generates a warning message when the temperature is below the acceptable value.
HiWarn		Displayed only with the `-v` and `-w` options. Generates a warning message when the temperature is above the acceptable value.
Min		Displayed only with the `-l` option. Value below Min threshold (warning condition).
Max		Displayed only with the -l option. The value exceeded Max threshold (fault condition).

See Also

Example

The `showenvironment` command display for the platform shell returns information on the slot number, device, sensor, value, units, age, and status, as shown in CODE EXAMPLE 2-58.

CODE EXAMPLE 2-58 `showenvironment` Platform Shell and Domain Shell Example

```
schostname:SC> showenvironment
Slot    Device   Sensor     Value  Units            Age     Status
----    ------   ------     -----  -----            ---     ------
SSC0    SBBC 0   Temp. 0    48     Degrees C        7 sec   ok
.
.
SB0     Board 0  1.5 VDC    0      1.51 Volts DC    1 sec   ok
.
.
/NO/SB2 Board 0  3.3 VDC    0      3.27 Volts DC    1 sec   ok
/NO/SB2 SDC 0    Temp. 0    49     Degrees C        1 sec   ok
.
.
/NO/IB6 Board 0  Temp. 0    28     Degrees C        9 sec   ok
.
.
RP0     Board 0  Temp. 0    26     Degrees C        11 sec  ok
.
.
PS0     48 VDC 0 Current 0  N/A    9.63 Amps        2 sec   ok
PS0     48 VDC 0 48 VDC 0   N/A    56.33 Volts DC   2 sec   ok
PS0     48 VDC 0 Temp. 0    N/A    37 Degrees C     2 sec   ok
.
.
IB7     Board 0  Temp. 0    27     Degrees C        13 sec  ok
```

showkeyswitch

Displays the keyswitch setting.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
showkeyswitch [-v] -d domainID
showkeyswitch -h
```

Syntax for the Domain Shell

```
showkeyswitch [-v]
showkeyswitch -h
```

Options / Parameters

`-h` displays help for this command.

`-v` is verbose mode.

`-d` *domainID* (platform shell only) is `a`, `b`, `c`, or `d`.

See Also

`setkeyswitch`

Description

Displays the keyswitch setting.

Examples

CODE EXAMPLE 2-59 `showkeyswitch` Example Showing the Keyswitch Set to On

```
schostname:A> showkeyswitch
key switch is: on
```

CODE EXAMPLE 2-60 `showkeyswitch` Example Showing the Keyswitch Set to Off

```
schostname:A> showkeyswitch
key switch is: off
```

CODE EXAMPLE 2-61 `showkeyswitch` Example Showing the Keyswitch Set to Standby

```
schostname:A> showkeyswitch
key switch is: standby
```

showlogs

Displays the system controller logged events stored in the system controller message buffer.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
showlogs [-d domain ID] [-v]
showlogs -h
```

Syntax for the Domain Shell

```
showlogs [-v]
showlogs -h
```

Options / Parameters

`-h` displays help for this command.

`-d` *domainID* (platform shell only) specifies a domain (`a`, `b`, `c`, or `d`).

`-v` is verbose.

Description

Displays the system controller logged events stored in the system controller message buffer. There is a specific log for each domain and the platform shell. If the loghost has been configured with `setupplatform` or `setupdomain`, messages will also be logged to the syslog host.

See Also

`setupdomain`

Examples

The examples for this command are very verbose and difficult to incorporate. Therefore, examples are not included.

showplatform

This command displays all configuration variables for this platform.

Scope

platform shell

Syntax

```
showplatform [-v] [-d domainID] [-p part]
showplatform -h
```

Options / Parameters

`-h` displays help for this command.

`-d` *domainID* specifies a domain to show (a, b, c, or d).

`-p` *part* displays information on a specific part.

where:

- `acls` shows the Access Control List (ACL).
- `loghosts` shows log hosts.
- `mac` shows the mac addresses, hostID, and the platform a serial number
- `network` displays the network settings.
- `partition` shows the partition mode.
- `sc` shows the system controller configuration.
- `snmp` shows snmp information.
- `status` displays domain status

-v is verbose mode.

Description

This command displays all configuration variables for this platform, including:

- Network attributes of the platform
- ACLs for each domain
- Domain status
- SNMP information
- Loghosts
- Partition configuration
- Mac address, host ID, and newly added system serial number

See Also

`setupplatform`, `showdate`

Example

CODE EXAMPLE 2-62 `showplatform` Output for a Sun Fire 6800 System

```
schostname:SC> showplatform

schostname:SC> showplatform

Domain  Solaris Nodename  Domain Status  Keyswitch
------  ----------------  -------------  ---------
A       -                 Powered Off    off
B       -                 Powered Off    standby
C       -                 Powered Off    off
D       -                 Powered Off    off

The system controller is configured to be on a network.
Network settings: DHCP
Hostname: hostname
IP Address: xxx.xxx.xxx.xx
Netmask: xxx.xxx.xxx.x
Gateway: xxx.xxx.xxx.xx0
DNS Domain:
Primary DNS Server: xxx.xxx.x.xxx
Secondary DNS Server: xxx.xxx.x.xxx

Loghost for Platform:
Loghost for Domain A:
```

CODE EXAMPLE 2-62 showplatform Output for a Sun Fire 6800 System *(Continued)*

```
Loghost for Domain B:
Loghost for Domain C:
Loghost for Domain D:

SNMP Agent: enabled
Chassis Description: Sun Fire 6800
Chassis Contact:
Chassis Location:
Trap Hosts:
Public Community String: P-public
Private Community String: P-private

ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain B: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain C: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain D: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

SC POST diag Level: off

Chassis is in dual partition mode.

schostname:SC>
```

CODE EXAMPLE 2-63 shows the showplatform -v command output.

CODE EXAMPLE 2-63 showplatform -v Output for a Sun Fire 6800 System

```
schostname:SC> showplatform -v

schostname:SC> showplatform

Domain  Solaris Nodename  Domain Status  Keyswitch
------  ----------------  -------------  ---------
A       -                 Powered Off    off
B       -                 Powered Off    standby
C       -                 Powered Off    off
D       -                 Powered Off    off

Network
-------
The system controller is configured to be on a network.
Network settings: DHCP
Hostname: hostname
IP Address: xxx.xxx.xxx.xx
Netmask: xxx.xxx.xxx.x
```

CODE EXAMPLE 2-63 `showplatform -v` Output for a Sun Fire 6800 System *(Continued)*

```
Gateway: xxx.xxx.xxx.xx0
DNS Domain:
Primary DNS Server: xxx.xxx.x.xxx
Secondary DNS Server: xxx.xxx.x.xxx

                          MAC Address          HostID
                          -----------          ------
Domain A                  xx:xx:xx:xx:xx:xx    xxxxxxxx
Domain B                  xx:xx:xx:xx:xx:xx    xxxxxxxx
Domain C                  xx:xx:xx:xx:xx:xx    xxxxxxxx
Domain D                  xx:xx:xx:xx:xx:xx    xxxxxxxx
SSC0                      xx:xx:xx:xx:xx:xx    xxxxxxxx
SSC1                      xx:xx:xx:xx:xx:xx    xxxxxxxx

Platform Serial Number: xxxxxxxx

Loghosts
--------
Loghost for Platform:
Loghost for Domain A:
Loghost for Domain B:
Loghost for Domain C:
Loghost for Domain D:

SNMP
----
SNMP Agent: enabled
Chassis Description: Sun Fire 6800
Chassis Contact:
Chassis Location:
Trap Hosts:
Public Community String: P-public
Private Community String: P-private

SNMP packets received: x
SNMP packets sent: xxx
SNMP traps sent: xxx

ACLs
----
ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain B: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain C: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain D: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

SC POST
```

CODE EXAMPLE 2-63 `showplatform -v` Output for a Sun Fire 6800 System *(Continued)*

```
-----
SC POST diag Level: off

Partition Mode
--------------
Chassis is in dual partition mode.

schostname:SC>
```

TABLE 2-29 Definitions of Headings for Status of All Domains in CODE EXAMPLE 2-62

Headers for CODE EXAMPLE 2-62	**Description**
Domain	Displays domains A through D. See domain in the Glossary.
Solaris Nodename	Contains the nodename of the domain (if the Solaris operating environment is running or has run since the last reboot of the system controller). Displays a dash (–) if the host name is not available.
Domain Status	Displays the domain state. The major values are: Powered off, Active-Solaris, Active-OpenBoot PROM, Active-Booting, Running POST, Standby, and Powered off.
Keyswitch	Displays the keyswitch position (off, standby, on, diag, secure).

showsc

Displays the uptime and version information about the system controller.

Scope

platform shell

Syntax

```
showsc [-h][-v]
```

Options / Parameters

`-h` displays help for this command.

`-v` is verbose mode.

Description

Displays the version and uptime information for the system controller.

See Also

Example

CODE EXAMPLE 2-64 showsc Command

```
hostname:SC> showsc
SC date: Wed Sep 06 05:45:57 PDT 2001
SC uptime: 29 seconds

ScApp version: 5.11.3
RTOS version: 17
```

RTOS is the Real Time Operating System for the system controller.

testboard

Tests the CPU/Memory board name you specify at the command line.

Scope

platform shell, domain shell

Syntax

`testboard` *board_name*

`testboard -h`

Options/Parameters

`-h` displays help for this command.

board_name for the CPU/Memory board is `SB0` – `SB5`.

Description

Tests the CPU/Memory board name you specify at the command line. The board status must be either Available or Assigned and the board must be powered on. To display the board status, use the `showboards` command. For more information on board status, see "Board States" on page 19. In order for the `testboard` command to run, the board in a domain must be assigned to the domain and/or be in the Access Control List (ACL). Also, sufficient Repeater boards must be powered on.

The `testboard` command runs tests at the levels set in the domain. Testing a CPU/Memory board is most accurately done when it is part of a domain. It is best if the board can be tested by POST. However, if that is not possible, testing a CPU/Memory board in the domain is the next best test.

Testing a CPU/Memory board in a domain has a very slight possibility to cause problems in the domain. If that is the case, test the board in the platform shell in order to keep it isolated from the domain.

See Also

`showcomponent, showboards`

Example

To test CPU / Memory board, `sb0` in the domain A shell, type:

TABLE 2-30 `testboard` Example Testing CPU / Memory Board 0

schostname:A> **`testboard sb0`**

Glossary

domain shell With the domain shell, you have access to system controller commands that you need to perform on a domain. There are up to four domain shells (A through D). The domain shell prompt is *schostname*`:A>` (or `B>`, `C>`, or `D>`.

domain A set of one or more system boards that act as a separate system capable of booting the operating system and running independently of other domains. Domains do not depend on each other and do not interact with each other.

domain console If the Solaris operating environment or the OpenBoot PROM is running, you can access the domain console. With the domain console, you can have the `ok`, `login`, #, or `%` prompts.

environmental monitoring All systems have a large number of sensors that monitor temperature, voltage, and current. The system controller polls devices in a timely manner and makes the environmental data available. The system controller will shut down various components to prevent damage.

keyswitch See virtual domain keyswitch.

partition A physical and logical separation of domains. You can set up the system with one partition or two partitions using the system controller `setupplatform` command. Partitions do not share Repeater boards

platform shell The platform shell enables access to the entire system and provides: configuration control, environmental status, ability to rearrange domains, ability to power on and off power grids, ability to change the system controller password, and other generic system controller functions.

POST Power-on self-test. This is the program that takes uninitialized system hardware and probes and tests its components, configures what seems worthwhile into a coherent initialized system, and hands it off to the OpenBoot PROM.

SRAM Static Random Access Memory. A type of high-speed memory device used for Ecache modules.

system controller The system controller consists of the System Controller board and the system controller software. The system controller provides communication pathways for console traffic and other data that needs to be passed between the system controller and the system. The system controller software monitors and controls the system, manages hardware, provides clock failover, manages domain consoles, and configures domains.

System Controller board A board containing a CPU (central processing unit), which oversees operation of the system and provides clocks and the console bus. These systems support two System Controller boards.

TOD See virtual time of day (TOD).

virtual domain keyswitch The system controller provides a virtual keyswitch for each domain. The `setkeyswitch` command controls the position of the virtual keyswitch for each domain.

virtual time of day (TOD) The TOD / NVRAM chip is located on the System Controller board. The system controller provides TOD services for each domain and for the system controller.

Index

A

active board state, 19
`addboard` command, 26
adding
  board to a domain, 26
assigned board state, 19
available board state, 19

B

board state
  Active, 19
  Assigned, 19
  Available, 19
board states, 19
boards
  CPU / Memory
    testing, 121
  listing assignment information and status, 95
  powering off, 61, 62, 65
  powering on, 64
  status, 95
  unassigning, 35
Break character, 28
`break` command, 28
break shell
  commands, 21
  exiting, 74

C

command history, 58
command summary, 21
console port
  commands, 21
CPU / Memory board
  testing, 121
current logging configuration, displaying, 113

D

`deleteboard` command, 35
displaying, 113
  current environmental status, 107, 108
  current logging configuration, 113
  version and uptime information, 120
domain
  A, entering from the platform shell, 15
  console
    definition, 13
  running the Solaris operating environment, 13
domain console
  resuming access to, 74
domain parameters
  setting up, 85
domain shell
  navigating to the OpenBoot PROM, 12
  navigating to the Solaris operating
    environment, 12

domains
resetting, 70

E

environmental status
displaying, 107, 108

F

flash PROMs, updating, 50
`flashupdate` command, 50

H

`help` command, 56
`history` command, 58

K

`keyswitch` command, 83
keyswitch positions, virtual, 83

L

listing board information, 95
logging configuration, current, 113

N

navigation
between domain shell and the OpenBoot PROM or the domain shell and the Solaris operating environment, 12
to the domain shell, 12

P

parameters
domain, setting up, 85
`password` command, 59
password, setting, 59
platform shell
entering domain A, 15
power grids
powering off, 61, 62
powering on, 64, 65
power supplies
powering off, 61, 62
powering on, 64, 65
powering off
power grids, power supplies, or boards, 62
powering on
power grids, power supplies, or boards, 65
`powern` command, 64
`poweroff` command, 61, 62
`poweron` command, 65

R

`reset` command, 70
resetting
domains, 70
`resume` command, 74

S

`setdate` command, 75, 77
setting, time, 75, 77
`setupdomain` command, 85
`showboards` command, 95
`showenvironment` command, 107, 108
`showlogs`, 113
`showsc` command, 120
Solaris operating environment, 12
status, boards, 95
status, environmental
displaying, 107, 108
system controller software
command summary, 21

T

`testboard` command, 121
testing
  CPU / Memory board, 121
time
  setting, 75, 77

U

unassigning a board from the domain, 35
updating the flash PROM, 50
uptime and version information, displaying, 120

V

version and uptime information, displaying, 120
virtual
  keyswitch, 83